Algorithm For Cross-Link Insertion To Improve Efficiency of Local Resonant Clock Trees

Edward Sullivan
Clemson University
Clemson, SC
elsulli@g.clemson.edu

Jas Condley
University of California, Santa Cruz
Santa Cruz, CA
jas@soe.ucsc.edu

Matthew Guthaus
University of California, Santa Cruz
Santa Cruz, CA
mrg@soe.ucsc.edu

***Abstract*—Resonant inductor-capacitor (LC) tanks in local clock distribution networks can reduce power consumption by 75% or more but are restricted to small local sectors due to high wire resistivity. The goal of this research is to further reduce power consumption and to increase local sector sizes by selectively inserting cross-links between branches of local trees while maintaining a low total capacitance. We present an algorithm to improve local resonant efficiency in terms of power delivered from the local source buffer to the local sector sinks. The algorithm evaluates candidate cross-links using criteria including link length and the distances from the link's source and destination nodes to the tank and to the sinks. Spice simulations demonstrate that on large enough trees, more than 50 % reduction in buffer power requirements is possible.**

Keywords-VLSI CAD; Non-tree Clocks; Cross-Links; Power-Efficiency

I. INTRODUCTION

Power reduction in Clock Distribution Networks (CDN) is a major concern of VLSI research because CDNs account for between 30 % and 70 % of chip power requirements, which is due to their large capacitances that are charged every clock cycle [1]. While CDNs are often organized hierarchically into global, regional, and local networks, the local networks are responsible for 70-90 % of the CDN's power requirements[1].

Inserting resonant inductor-capacitor (LC) tanks, tuned to the frequency of the clock cycle, into local trees can improve tree power efficiency by 75 % or more. The tank is positioned in the 10 % of the tree closest to the root at a node for which the average sink voltage swing is the greatest. In contrast to nonresonant local trees which may have several buffers placed throughout the tree to prevent signal attenuation, resonant trees rely on a local source buffer placed at the root so as to allow the uninhibited propagation of a single resonant signal through all parts of the tree. The drawback of this design is that the LC tank, specifically the inductor, requires a lot of space and only supports a few sinks because the tree does not have multiple buffers to prevent signal attenuation in the wires.

The aim of this research is to use cross-link insertion to further improve the efficiency of local resonant trees so that the sector size may be increased and fewer LC tanks are required in the CDN. Previous algorithms involving cross-link insertion have dealt with nonresonant CDNs and have focused on skew-reduction with minimal increase in tree power [2].

The theory behind the proposed algorithm relies on the well-known concept that the equivalent resistance of multiple resistors in parallel between two nodes is less than the resistance of any of the individual resistors of the group. Yet while the lowered resistance may reduce power dissipation in wires, the additional horizontal area of cross-links increases the network's capacitance proportionally, causing the CDN to behave more like a low-pass filter and induce slew. Therefore, in the algorithm presented here, a limit was set on the total allowed percent increase in nontree capacitance.

Similarly, an algorithm based on widening wires could be employed to reduce a tree's resistance because a wire's resistance is inversely proportional to its cross-sectional area. Yet given the restraints on the permissible increase of the CDN's capacitance, an algorithm based on wire widening likely has less potential for effectiveness than one based on inserting cross-links. Whereas the reduction in resistance between two nodes of a tree resulting from the widening wires is strictly inversely proportional to the increase in capacitance of the wires, the insertion of a cross-link in a tree will achieve a better ratio of resistance reduction per added capacitance if the cross-link is shorter than the existing wire path between the two nodes. Therefore, an algorithm based on Cross-Link Insertion was developed to reduce clock network energy requirements.

II. CROSS-LINK INSERTION ALGORITHM

A. Preliminary Observations

In general, shorter links were more effective at reducing tree power requirements without increasing tree capacitance because shorter links have proportionally less resistance and capacitance. Furthermore, most links were only effective if used to connect a node near the tank (often in the center of the tree) to a node near a sink. That is, if two nodes were equal distance from the tank or equal distance from the nearest common ancestor node, a link between those two nodes would likely be ineffective. Links closer to the tank were usually more effective than links farther away, even when the links' destination nodes did not have more sinks in their fan-outs, though no explication is presented here for this trend. These observations were used to formulate the Cross-Link Insertion Algorithm.

B. *Pre-conditions*

Before applying the Cross-Link Insertion Algorithm, the Deferred-Merge Embedding algorithm (DME) was used to determine the layout of the tree and the LC tank was placed.

Also, before applying the Cross-Link Insertion Algorithm, the number of nodes on the tree was increased by segmenting into two equal parts each wire longer than four times the original average wire segment length until no wire segment on the tree was longer than four times the original average wire segment length. The reason for increasing the number of nodes on the tree was to increase the number of potential cross-links because during the Cross-Link Insertion Algorithm presented here, cross-links are only connected between two nodes and are never connected to any non-nodal locations along the length of a wire. Without segmenting various long wires on the tree, the only nodes that are available are those at corners and branching points. While this optional step may improve tree efficiency by allowing a larger pool of potential nodes to use for cross-link connections, increasing the number of nodes on the tree may increase the run-time of the algorithm.

C. *Method*

The algorithm below was developed using theory and heuristic observations. Various coefficients and threshold percentages may be adjusted depending on the specific topological characteristics of the particular tree (i.e., density of sinks, blockages) and the desired runtime of the algorithm.

Links that directly overlapped other wires or links were allowed because such links could be implemented physically by slightly adjusting the link's position so as to prevent the overlapping and the adjustment would cause only a negligible change to the tree's topology.

1. To reduce the run-time of the algorithm and exclude links likely to be ineffective, all the nodes in the tree were examined and only if a particular node was a certain minimum wire distance away from every sink in the tree, was it then added to a narrowed list of potential source nodes. A link's source node was defined as the node on the existing tree that the link attaches to and which is closest to the tank. The link's destination node was defined as the node further from the tank. The threshold value used was 75% of the distance from the tank to its closest sink. Also, any node on the path between the tank and root (not including the tank node) which attached to fewer than three wire segments was excluded from the list because cross-links involving these nodes never improved the efficiency of the tree, perhaps because such nodes diverted current away from the tank.

2. For each source node, all the other tree nodes within a certain Manhattan distance were considered as destination nodes for potential links and a list of potential links was generated. A potential destination node was excluded if it was farther away from the source node in Manhattan distance than 50 % of the wire length from the tank to the sink which was the farthest wire length away from the tank. This restriction reduced the run-time of the algorithm by excluding links that would likely be too long to be effective. As before, any node between the tank and root (not including the tank node) which attached to fewer than three wire segments was not considered as a potential destination node. Furthermore, no repeat links were allowed, though in some cases, repeat links could have been beneficial. If for any link the destination node was closer to the tank than the source node, the link was reversed so that the source node would be the one closer to the tank.

3. Each potential link was given a score *S* according to the following formula. The scoring scale was designed with the intent that the links which could cause the greatest reduction in tree power requirements per percent increase in tree capacitance would receive the highest scores.

$$S = \frac{A^{1.5} \cdot B^{1.5}}{(C+D+E)^{1.5} \cdot D^{1.5} \cdot E^{1.5}} \qquad (1)$$

- In (1), *A* represents the absolute value of the difference between the wire lengths from the link's source and destination nodes to the respective sinks which are the shortest wire length away.
- *B* is the absolute value of the difference between the wire lengths from the link's source and destination nodes to the LC tank.
- *C* is the wire length from the source node to the tank.
- *D* is the wire length from the destination node to the tank.
- *E* is the wire length of the potential link (i.e., the Manhattan distance between the source and destination nodes).

4. All the potential cross-links were sorted according to their Scores determined by (1). Links were then added to the tree in order of highest *S* value until no link could be added to the tree without exceeding the allowed percent increase in tree capacitance, here tested at 5 % and 10 %. No links were added twice, although in some cases, redundant links may have been more effective than nonredundant links at reducing power. A potential link was also ignored if it had a destination node too close in wire length to the destination node of a link that had already been added. The purpose of this rule was to prevent the clustering of redundant links in one section of the tree and this rule helped improve tree efficiency while reducing tree complexity. The minimum allowed distance

between link destination nodes was 10 % of the average wire length from the tank to each sink. Even if the capacitance limit had not been reached, the link insertion process was stopped if the remaining available links to insert were in the lowest 25 percentile of potential cross-links as ranked by Score.

D. Post-conditions

After adding cross-links, the inductor in the LC tank should be resized because the apparent capacitance of the tree at the tank may have been changed. However, the tank generally does not need to be repositioned because the addition of cross-links should reinforce the advantage of the original tank location.

III. Experimental Parameters

Thirty-six benchmarks of varying sector size (65,000 nm to 100,000 nm) were tested at 3,500 Hz. Each sector was generated using random sink placement and each sector had the same density of sinks, about 1 sink per 10^8 sq. nm. Each sink had a random capacitance in the fempto Farad range. Every benchmark was tested with the Cross-Link Insertion Algorithm applied with a 10% allowed increase in tree capacitance, applied with a 5% allowed increase in tree capacitance, and not applied at all. Ng-Spice 21 was used to measure the trees' power values. For every test, measurements were taken after the algorithm had been applied (or not applied) and the inductor had been resized. No new tank location was searched for after adding the cross-links and the local source buffer, which had been set to the smallest size, was never resized. The average of sink average power was determined by measuring the average current and voltage before each sink in the tree over one period after 19 periods had already elapsed.

Each benchmark had two buffers at the root, one to represent the incoming signal to the network and one, closer to the sinks and used to drive the network, called the local source buffer. The former was identical for every benchmark while the latter varied by the specific network's power needs. Measurements of the network's required buffer power refer in this experiment only to the local source buffer.

The results for one benchmark (sector size 86,000 nm) were ignored because when the benchmark was tested without applying the Cross-Link Insertion Algorithm, the second sizing of the inductor significantly changed the tree's power characteristics, though ideally the size of the inductor and the tree power information would not have been changed since no cross-links had been added.

IV. Results

Fig. 1 shows that the Cross-Link Insertion Algorithm, when used with a 10 % limit on capacitance increase, can often improve tree efficiency by between 30 % and 60%.

The algorithm was most effective at reducing the local source buffer power requirements when the initial tree capacitance was large. The correlation between buffer power reduction and initial tree capacitance may result from the fact that networks with more capacitance have more total wire length, more nodes, and thus a larger number of potential cross-links. Also, trees with more wire length may perhaps have on average longer wires with greater resistances, which have more potential for reduction in resistance. Although larger sectors generally have greater total capacitances, the correlation between reduction in buffer power requirements and sector size (Fig. 2) was much weaker than the correlation between buffer power requirements and total capacitance.

When the algorithm was tested with a 5 % allowance on nontree capacitance increase (Fig. 3), the algorithm still successfully reduced local source buffer power requirements, though on average, it was 7% less effective than when the 10 % capacitance allowance was used (Table I).

Even though the local source buffer power requirements were successfully reduced in each test, the average power delivered to sinks in each test also improved. The average increase was 4.08 % when a 5 % capacitance increase allowance was used and was 3.03 % when a 10 % capacitance increase allowance was used (Fig. 4). Similarly, the average power of the weakest power sink was improved in each test.

A drawback of the algorithm was that, although the average sink voltage-swing was improved in all but one test, the minimum sink voltage swing was decreased in 22 of the 35 tests when a 5% capacitance increase allowance was used and in 25 tests when a 10% capacitance increase allowance was used. As listed in Table I, the average voltage-swing of sinks increased by an average of 4.60 % and 4.73 % when 5 % and 10 % capacitance increase allowances were used, respectively, but the minimum sink voltage swings decreased by an average of 1.94 % and 2.92 %.

The percent change in skew varied from benchmark to benchmark, but was overall neutral, as reported in Fig. 6 and Table I.

Figure 1. The Cross-Link Insertion Algorithm (10 % capacitance limit) was more effective at reducing local source buffer power requirements on trees with greater initial capacitances

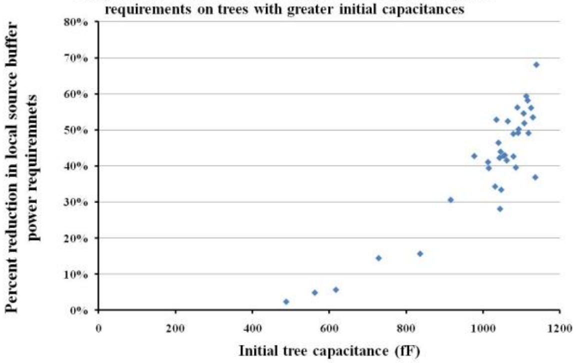


Figure 2. Little correlation between sector size and effectiveness of the Cross-Link Insertion Algorithm (10 % capacitance limit)

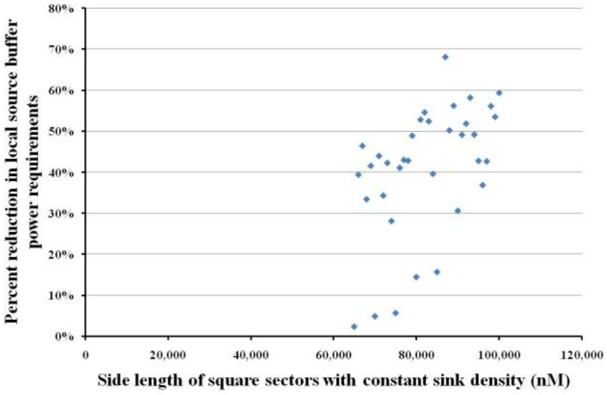


Figure 3. The Cross-Link Insertion Algorithm with a 5 % capacitance increase limit was about 7 % less effective at reducing buffer power needs than with a 10 % capacitance increase limit

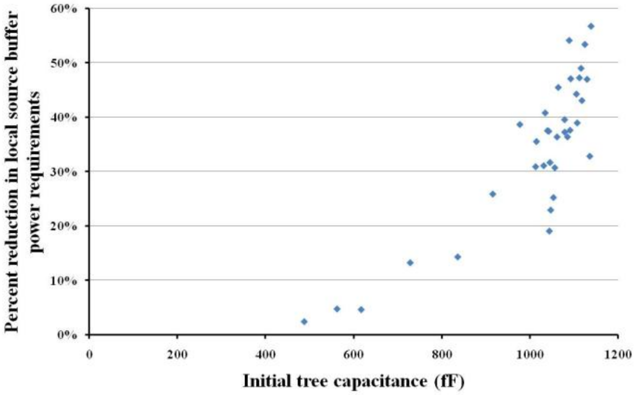


Figure 4: The Cross-Link Insertion Algorithm (10 % capacitance increase limit) improved power delivered to each sink

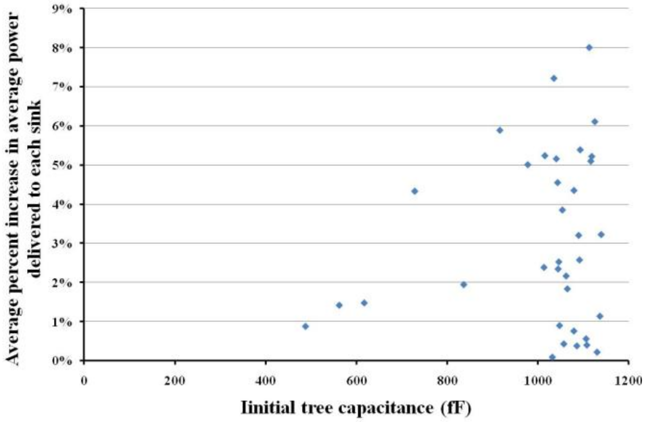


TABLE I. AVERAGE VALUES FOR THE CROSS-LINK INSERTION ALGORITH TESTED WITH VARYING CAPACITANCE INCREASE ALLOWANCES

Value averaged over 34 benchmarks	*5 % Capacitance Increase Allowance*	*10 % Capacitance Increase Allowance*
Required local source buffer power	-34.03 %	-40.94 %
Change in average sink voltage swing	4.60 %	4.73 %
Change in minimum sink voltage swing	-1.94 %	-2.92 %
Change in average sink power	4.08 %	3.03 %
Change in minimum sink power	4.18 %	3.17 %
Skew	6.96 %	-6.93 %

V. CONCLUSION

The Cross-Link Insertion Algorithm presented here demonstrates that selectively placed cross-links can reduce local source buffer power requirements by more than 50 % while improving average power delivered to sinks and voltage-swing. The algorithm, based solely on the tree's topology and the placement of the resonant LC tank, appears to scale well to trees with larger capacitances. Future study on this topic may address the challenge of ensuring that the voltage-swing at each sink never drops below a threshold value necessary for proper operation of the sink device.

Figure 5. The Cross-Link Insertion Algorithm (10 % capacitance limit) improved sink voltage-swing as initial tree capacitance increasesd

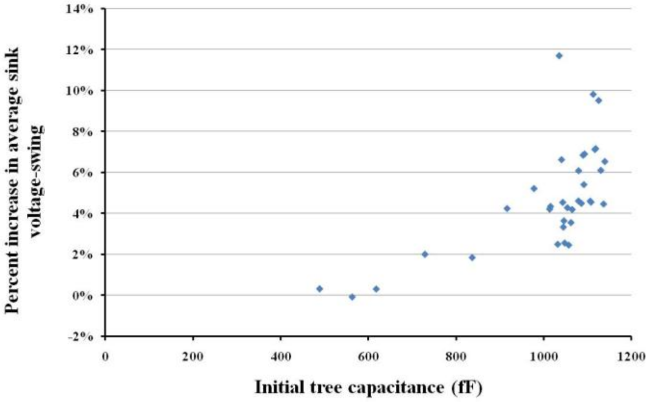
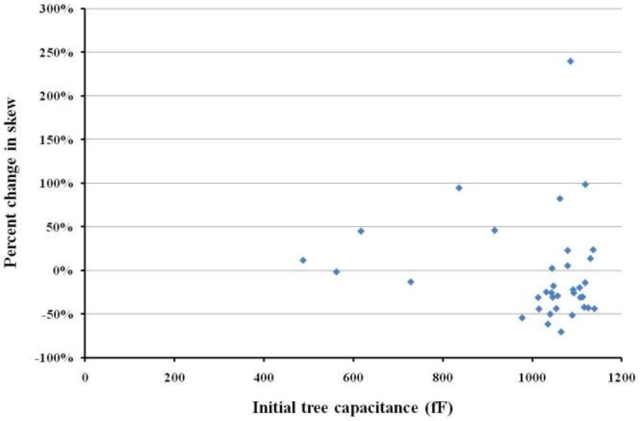


Figure 6. The Cross-Link Insertion Algorithm (10 % capacitance increase limit) on average did not worsen skew greatly



ACKNOWLEDGEMENT

This work was supported by the UCSC SURF-IT 2010 Research Experiences for Undergraduates Site, NSF grant CNS-0852099, <surf-it.soe.ucsc.edu>. We would also like to acknowledge the members of the VLSI-DA group at UCSC for their guidance and support. Last but not least, we would like to thank the staff and faculty at UCSC who contributed to the SURF-IT program.

REFERENCES

[1] M. R. Guthaus, G. Wilke, and R. Reis. Revisiting automated synthesis of high-performance clock networks. *In IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2011.

[2] A. Rajaram, D. Z. Pan, and J. Hu. Improved algorithms for link-based non-tree clock networks for skew variability reduction. In *Proceedings of the ISPD*, San Francisco, CA, April 2005, pages 55–62.