

Sensitivity Based Link Insertion for Variation Tolerant Clock Network Synthesis

Joon-Sung Yang[†], Anand Rajaram^{†‡}, Ninghy Shi[†], Jian Chen[†], and David Z. Pan[†]

[†] Dept. of ECE, University of Texas, Austin, TX 78712

[‡] Dallas DSP Design, Texas Instruments Inc., Dallas, TX 75243

Abstract

Clock distribution is one of the key limiting factors in any high speed, sub-100nm VLSI design. Unwanted clock skews, caused by variation effects like manufacturing variations, power-ground noise etc., consume increasing proportion of the clock cycle. Thus, reducing the clock skew variations is one of the most important objectives of any high-speed clock distribution methodology. Inserting cross-links in a given clock tree is one way to reduce unwanted clock skew variations [1–6]. However, most of the existing methods like [1–5] use empirical methods and do not use delay/skew variation information to select the links to be inserted. This can result in ineffective links being inserted. The work of [6] considers the delay variation directly, but it is very slow even for small clock trees. In this paper, we propose a fast link insertion algorithm that considers the delay variation information directly during link selection process. Our algorithm inserts links only in the parts of the clock tree that are most susceptible to variation effects by evaluating the skew sensitivity to variations. Another key feature of our algorithm is that it is compatible with any higher order delay model/variation model, unlike the existing algorithms. We verify the effectiveness of our algorithm using HSPICE based Monte Carlo simulations on a set of standard benchmarks.

1. INTRODUCTION

Clock network synthesis is one of the most important steps in any high-speed digital VLSI design. Since clock network is typically the largest net in any design, it is more sensitive to variations than regular signal nets. Some of the dominant variation effects that affect the clock network are manufacturing variations [7–9], temperature variation [10] and power supply variations [11]. It is to be noted here that the skew variation can be a very significant portion of the *nominal* skew. For example, according to [9], just the interconnect variations can contribute to as much as 25% skew variation. Such a significant impact makes the traditional method of margining for variation both inefficient and risky. Thus, designing variation tolerant clock networks is of utmost importance for today's high-speed designs.

Several methods have been proposed for designing variation tolerant clock networks. Among these, the link based clock

network [1] is a promising approach that combines the advantages of clock mesh and clock trees. The basic idea of [1] is to insert links in appropriate locations in a given clock tree, there by changing the tree to a non-tree structure. Due to its redundancy, the non-tree structure is much more tolerant to variation effects compared to a tree structure. The key to building a good link based non-tree clock network is selection of appropriate link locations. Though the current link selection algorithms of [1–5] result in significant skew reduction, they suffer from a number of drawbacks which are discussed in Section 2-B. The important contributions of this work are:

- We introduce the concept of delay sensitivity vector for the sink pairs. We use this to identify the sink pairs that are most susceptible to variation effects and insert links only between such sink pairs. This makes our algorithm very efficient and results in a robust, variation tolerant clock network.
- Our methodology is compatible to higher order delay models. Though we use Elmore delay in this paper to explain our algorithm, the same methodology can be easily extended to any higher order delay model.
- Any given variation model, like the systematic and random power supply variation, temperature variation etc. can be seamlessly integrated with our algorithm. In this work, we use random interconnect width variation as an example.

SPICE based Monte Carlo simulations on standard benchmarks shows that our algorithm achieves better skew variability reduction while utilizing considerably lesser routing resources compared with existing methods.

2. REVIEW OF EXISTING WORKS

According to [1], if a link is inserted between nodes u and w of a clock tree, the final skew between nodes u and w after link insertion is given by:

$$\hat{q}_{u,w} = \frac{R_l}{R_l + r_u - r_w} (q_{u,w} + \frac{C_l}{2} (R_{u,u} - R_{w,w})) \quad (1)$$

where, $q_{u,w}$ is the original skew between nodes u & w , $\hat{q}_{u,w}$ the final skew after the link insertion, C_l the link capacitance, R_l the link resistance, $R_{u,u}$ & $R_{w,w}$ the transfer resistances of nodes u & w . The r_u & r_w are equal to the Elmore delay at u & w when $C_u = 1$, $C_w = -1$ and the other node capacitance are

zero. The C_u and C_w are the total node capacitance at nodes u and w respectively. Equation 1 can be rewritten as:

$$\hat{q}_{u,w} = (\alpha q_{u,w} + \alpha\beta) \quad (2)$$

where $\alpha = \frac{R_l}{R_l + r_u - r_w}$ and $\beta = \frac{C_l}{2}(R_{u,u} - R_{w,w})$.

According to [1], the value of α is always less than 1. The first term of the above equation is the value of the original skew scaled by α . The second term represents the extra skew introduced by the value of link capacitance. If the original skew between nodes u and v is zero, then we can conclude that adding *only* a resistance still maintains the zero skew. However, adding the link capacitance changes the nominal skew. In [1], the original skew after link insertion is restored by tuning the locations of the internal clock network nodes similar to the method in [12]. This makes sure that the link insertion does not affect the nominal skew of the clock network. It also guarantees that link insertion between sinks u and w always reduces the skew as shown in [1]. Also, it has been proved in [1] that adding the links lower in the clock tree is better for skew variability reduction. Henceforth, we will assume that the links will be inserted only between clock sinks in this work.

All the link insertion algorithms of [1–5] follow the steps listed below:

- Given an initial clock tree, select all the node pairs where cross links are to be inserted.
- Add only the link capacitance values at the sink nodes and tune the internal nodes of the clock tree to restore original skew. The tuning is done in a bottom-up fashion similar to the method in [12].
- Finally, add the link to the selected node pairs. Since the effect of link capacitance has been already removed by bottom-up tuning, only the effect of link resistances will be present.

Selecting the sink node pairs for link insertion is the critical step in the above procedure. All the previous works mainly differ in the way the links are chosen.

A. Existing Link Insertion Methods

The existing link insertion methods can be broadly classified into the following three types:

Rule Based Algorithms: In the rule-based link insertion methods of [1–5], all the possible links are characterized by parameters denoted by α , β , γ and δ . The definitions of α and β are the same as in equation 2. The readers are referred to the [1–5] for a detailed explanation of all the parameter definitions. In simple terms, the rule-based methods involve empirically selecting appropriate values of bounds α_{max} , β_{max} , γ_{max} and δ_{max} and adding only the links that have α , β , γ and δ values less than chosen bounds.

Graph Theory Based Algorithms: The graph theory based link insertion methods of [1–5] involve appropriately dividing the clock tree into recursive bi-partite graphs and choosing the number of links to be inserted based on edge weights of the bi-partite graph. The edge weights are proportional to the wire-length of the links considered.

Statistical Link Insertion: The statistical link insertion method of [6] adds the links incrementally by updating the statistical values of the sink delays and using the information of the latest non-tree to select the next link to be inserted.

B. Drawbacks of the Existing Approaches

The key drawbacks of the existing algorithms are:

- Most current algorithms require the user to empirically select the values of different parameters. For rule-based methods, parameters like α_{max} , β_{max} , γ_{max} and δ_{max} are chosen. In the case of the graph theory based methods, the user needs to empirically select the number of links to be inserted at each level of the clock network.
- Most of the existing works, except [6], do not use either delay or skew variation information while choosing links. Therefore, they might add ineffective links.
- The current methods are not compatible with the use of higher order delay models because all the parameters used for selecting the links are based on nominal Elmore delays.
- Also, the current algorithms cannot be modified for use with a given variation model. For example, if the IR drop variation information or the metal thickness variation information is available, the current methods cannot use them to choose between links that have similar parameter values, but are different only because of their variation context.
- Though the method of [6] does not suffer from some of the above drawbacks, it has very high run-time and complexity. For example, the method takes 2 minutes to add 10 links in a 74 pin clock network and more than 10 hours to add 20 links to a 597 sink clock network.

3. SENSITIVITY BASED ALGORITHM

In this section, we propose the delay sensitivity based algorithm that addresses all the drawbacks of the existing algorithms. The basic idea behind sensitivity based link insertion is to insert links between the node pairs that are most susceptible to variation. This is motivated by the fact that a given link has the maximum beneficial effect on the node pairs between which it is inserted. The rest of this section presents the details of the algorithm. First, we introduce the concept of delay sensitivity vector for a given process variation model. This concept helps us to identify the link pairs that are the most susceptible to the variation effects. Following this, we prove that the α -rule proposed for unbuffered clock trees in [1] is also true for buffered clock tree case. Finally, we present our sensitivity based link selection method.

A. Sensitivity Vector

Consider Fig.1 which shows a typical clock tree. Let the variables x_1, x_2, \dots, x_n denote process variation variables that affect the delays and skew of the clock tree. Without any loss of generality, we can assume that these variables are independent from each other. If the variables are not independent from each other, then we can employ the method of [13] to create

independent process variation variables. Let T_i denote the root to sink delay of any sink i . This can be expressed in terms of the process variation variables x_1, x_2, \dots, x_n as follows:

$$T_i = f_i(x_0, x_1, x_2, \dots, x_n)$$

The first order Taylor expansion of T_i can be written as

$$\begin{aligned} \hat{T}_i &\approx T_{i,0} + \frac{\partial T_i}{\partial x_0} \Delta x_0 + \frac{\partial T_i}{\partial x_1} \Delta x_1 + \frac{\partial T_i}{\partial x_2} \Delta x_2 + \dots \\ &+ \frac{\partial T_i}{\partial x_i} \Delta x_i + \dots + \frac{\partial T_i}{\partial x_n} \Delta x_n \\ &= T_{i,0} + \sum_{k=0}^n \frac{\partial T_i}{\partial x_k} \Delta x_k \end{aligned}$$

where $T_{i,0}$ is the nominal delay of node i and $\Delta x_k = x_k - x_{k_nominal}(k \in (0 \dots n))$.

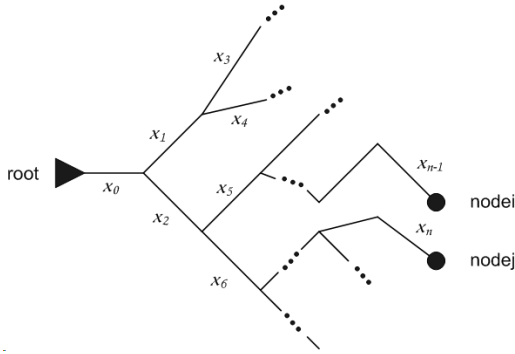


Fig. 1. Clock Tree with Variations on Each Segment

It may be noted here that, expressing the delay variation as a linear function of the random variables is quite accurate for all practical purposes [14]. In other words, though the delay might be a non-linear function of the variables x_i , the variation in delay w.r.t. its nominal value can be treated as a linear function of changes in the random variables.

Assuming a zero nominal skew clock tree, $T_{i,0}$ is equal to $T_{j,0}$ for any two sinks i and j . Therefore, the skew between sinks i and node j in the presence of process variation can be expressed as:

$$\begin{aligned} q_{i,j} &= T_i - T_j \approx \hat{T}_i - \hat{T}_j \\ &= \left[\Delta x_0 \Delta x_1 \Delta x_2 \dots \Delta x_n \right] \begin{bmatrix} \frac{\partial T_i}{\partial x_0} - \frac{\partial T_j}{\partial x_0} \\ \frac{\partial T_i}{\partial x_1} - \frac{\partial T_j}{\partial x_1} \\ \vdots \\ \frac{\partial T_i}{\partial x_n} - \frac{\partial T_j}{\partial x_n} \end{bmatrix} \end{aligned} \quad (3)$$

Let us define the sensitivity vector for the skew between nodes i and j as:

$$S_{i,j} = \begin{bmatrix} \frac{\partial T_i}{\partial x_0} - \frac{\partial T_j}{\partial x_0} \\ \frac{\partial T_i}{\partial x_1} - \frac{\partial T_j}{\partial x_1} \\ \vdots \\ \frac{\partial T_i}{\partial x_n} - \frac{\partial T_j}{\partial x_n} \end{bmatrix} \quad (4)$$

Let $M_{i,j}$ be the *Magnitude* of $S_{i,j}$, we have

$$M_{i,j} = \sqrt{\sum_{k=0}^n \left(\frac{\partial T_i}{\partial x_k} - \frac{\partial T_j}{\partial x_k} \right)^2}$$

Thus, the skew between any two sink nodes i and j can be expressed in terms of the sensitivity vector as:

$$q_{i,j} = \left[\Delta x_0 \Delta x_1 \Delta x_2 \dots \Delta x_n \right] S_{i,j}$$

An important point to be noted here is that each element of the vector $S_{i,j}$ in equation (4) represents effect of one process related variable on the skew between sink nodes i and j . Similar to the work of [6], we also assume that the variables have a Gaussian distribution. Based on this assumption, we can regard the Δx_k ($k = 0, 1, \dots, n$) as zero mean random variables with a known variance σ_k . Since a weighted sum of several independent Gaussian random variable is also Gaussian, the skew between sink nodes i and j , $q_{i,j}$ is also a Gaussian random variable. $q_{i,j}$ has a range of $E(q_{i,j}) - 3\sigma_{i,j} \sim E(q_{i,j}) + 3\sigma_{i,j}$, where the $\sigma_{i,j}$ is the standard deviation of the $q_{i,j}$. For a zero nominal skew clock tree, the expected value of skew is given as:

$$E(q_{i,j}) = \left(E \left(\sum_{k=0}^n \left(\frac{\partial T_i}{\partial x_k} - \frac{\partial T_j}{\partial x_k} \right) \Delta x_k \right) \right) = 0 \quad (5)$$

The range of $q_{i,j}$ is $-3\sigma_{i,j} \sim +3\sigma_{i,j}$. Similarly, the standard deviation of skew is calculated as

$$\sigma_{i,j} = \sqrt{\sum_{k=0}^n \left(\frac{\partial T_i}{\partial x_k} - \frac{\partial T_j}{\partial x_k} \right)^2 \sigma_k^2} \quad (6)$$

where σ_k is the standard deviation of the independent Gaussian distribution variable Δx_k . Assuming that Δx_k ($k = 0, 1, \dots, n$) have the same standard deviation σ , we can rewrite the above equation as

$$\sigma_{i,j} = \sqrt{\sum_{k=0}^n \left(\frac{\partial T_i}{\partial x_k} - \frac{\partial T_j}{\partial x_k} \right)^2} \cdot \sigma = M_{i,j} \cdot \sigma \quad (7)$$

The magnitude of standard deviation of skew between sinks i and j is indicative of how sensitive the skew between nodes i and j is to the variation effects. Therefore, we can conclude that the sink pairs that have higher value of $M_{i,j}$ are more sensitive to variation. Thus, by adding link only between sinks pairs that have the maximum value of $M_{i,j}$, we can greatly reduce the sensitivity to variations.

Obtaining Sensitivity Vector for a Generic Case: In the above derivation, we have assumed that the random variables Δx_k as Gaussian random variables with the same σ just for illustrative purpose. The concept of using the magnitude of sensitivity vector as a measure of skew sensitivity to variation is still applicable even if the random variables Δx_k had different non-Gaussian distributions.

Similarly, the same concept can also be applied with higher order delay models, including SPICE, by employing a method

similar to the work of [15] to obtain the delay sensitivities. For example, we can obtain the sensitivity vector considering the effect of power-supply noise as follows. For a given buffer in the clock tree, any changes in the power-supply voltage on that buffer directly affects buffer delay and the delay of the segment directly driven by it. Also, it can have an effect on the input slews of next stage, thereby affecting the delay of the next stage slightly. In most practical cases, the effect of slews on delays can be ignored after one stage of buffers. The key point is that, for all practical purposes, we can bound the effect of small changes in power-supply voltage on a given buffer to a few segments near the buffer. Thus, we need not evaluate the delay sensitivities of all the segments in the clock tree for a given power-supply noise in a single buffer as most segment sensitivities can be assumed to be zero for all practical purposes. By extension, we can state that we can obtain the delay sensitivities of all the segments of a given clock tree w.r.t. the power-supply noise for all the clock buffers in an efficient manner. Once we know the delay sensitivities of each segment of the clock network, we can trivially obtain the sensitivities of the sink delays and thereby the sensitivities of skew between any two sinks. This method can be similarly extended to consider variation effects like L_{eff} , T_{ox} , interconnect thickness, etc. Also, the above method of obtaining sensitivities inherently considers the path sharing between different sinks. For example, when a given segment is common for two sinks, its effect will be cancelled out while obtaining the values of $S_{i,j}$ in Equation 4.

B. α -Rule for Buffered Clock Tree

The α rule that was proposed in [1] was motivated by the fact that it scales down the original skew as shown in equation 2. Further, it was shown in [1] that the value of α is always less than 1 for any link added between nodes driven by the same buffer. However, it was not clear whether the same α rule could be extended for links that connect nodes driven by different clock buffers. This situation can happen in a buffered clock network. In this section, we prove that the same rule is also valid for buffered clock networks where a given link can have two drivers.

The motivation for extending the α -rule for buffered clock tree is explained below. The magnitude of sensitivity vector, $M_{i,j}$, is a good measure of how much a the skew between a given sink pairs i, j is susceptible to variation effects. However, the value of $M_{i,j}$ does not measure how close two sink pairs are in terms of physical proximity. For example, two sets of sink pairs might have the same value of $M_{i,j}$, but one of the sink pairs might be physically close to each other while the other pair might be far apart. In such cases, we need to be able to add link between the sink pair that are close to each other to reduce the wire-length consumption. From the work of [1], we can conclude that lengthy link will have a high value of α . Thus, if we are able to choose links with low value of α , we will be able to have a good control over the increase in wire-length due to link insertion.

It is worth noting here that equation 2 was derived in [1] based on the following equation from [16]:

$$\hat{\mathbf{v}} = \mathbf{v} - \frac{v_i - v_j}{R + r_i - r_j} \mathbf{r} \quad (8)$$

The above equation gives the new voltage in all the nodes of any RC network when a resistance R is added between nodes i and j of the RC network. The $\hat{\mathbf{v}}$ is the vector of new node voltages and \mathbf{v} is the original node voltages before adding the resistance R , v_i and v_j are the initial node voltages of i and j . The \mathbf{r} vector gives the Elmore delays of all the nodes of the RC network when the capacitance at node i is +1 and at node j is -1 and all other capacitance values set at 0. v_i and v_j are the entries in the \mathbf{r} for the nodes i and j .

The work of [16] also deals with situations when there are more than one voltage source by chopping up the network into sub-networks such that the each sub-network has only one source. After this, the different sub-networks are stitched together considering the fact that the final settling voltages of the different nodes at $t = \infty$ can be different from what it was when it was driven by only one source. However, in the case of a link driven by two different buffers, since the entire clock tree can be assumed to be on the same voltage domain, we can directly apply equation (8) for delay calculation purposes. As a result, the same α rule proposed in [1] for non-buffered clock trees is also valid for buffered clock trees. From [1], we know that lower the value of α , better the link is. Since the value of α is always less than 1, we can also state that a higher value of $(1 - \alpha)$ denotes a better link.

C. Sensitivity Based Link Insertion

Based on the last two sections, we can conclude that the effectiveness of a link between nodes i and j is proportional to $M_{i,j}$ and $(1-\alpha)$ where $M_{i,j}$ is the magnitude of the sensitivity vector and α is as defined in equation 2. Hence, we define the link sensitivity cost as follows:

$$SensitivityCost_{i,j} = (1 - \alpha_{i,j})M_{i,j} \quad (9)$$

Any node pair with large value of eq (9) can be regarded as a good candidate for link insertion. This is because if $M_{i,j}$ is high, it means that the sink pair i, j is very sensitive to variations. Similarly, if the value of α is low (or a high value of $1-\alpha$), the link can significantly reduce the final skew. Thus, we use the *SensitivityCost* as a measure of effectiveness of each link.

In this paper, as in [3], we follow the set of guidelines listed below for link insertion:

- Links are always inserted between zero nominal skew nodes.
- Node pairs are selected such that the buffers driving the links are not overloaded. Therefore, the selected sink nodes are relatively close to each other. This also makes sure that the links are not concentrated in a single place.

- Since the magnitude of the sensitivity vector is higher for sink pairs that do not share any common element, the method inherently selects pairs with considerably different source to sink paths.
- Short circuit risk in multi-driver nets is avoided by the method of [3].

The overall flow of the sensitivity based link insertion for buffered clock trees is shown below.

Algorithm 1 Node Pair Selection

Input: Buffered Clock Tree Node-List

- 1: Calculate delay sensitivity for each independent segment of the clock network
 - 2: **for** Each sink pair i, j **do**
 - 3: **if** Link distance $<$ Max link length AND no short circuit problem [3] **then**
 - 4: Generate sensitivity vector for node pair and obtain its magnitude $M_{i,j}$
 - 5: Compute $(1 - \alpha)M$
 - 6: $SensitivityCost_{i,j} = M_{i,j} \cdot (1 - \alpha)$
 - 7: **end if**
 - 8: **end for**
 - 9: Sort $(1 - \alpha)M$ and find the top N links such that the wire-length increase reaches limit.
 - 10: Add all link capacitance to the sink nodes
 - 11: Tune the locations of the internal nodes to restore original skew similar to [1]
 - 12: Add link resistances
 - 13: Output the final link-based non-tree clock network
-

As shown in the algorithm, we first obtain the delay sensitivity of each clock segment under the given variation model. This step enables us to quickly get the skew sensitivities for all the sink pairs. Next, for every feasible node pair, we evaluate the value of the *SensitivityCost*. Finally, we select the top N links based on the value of *SensitivityCost* such that the total wire-length does not exceed the user specified maximum wire-length increase.

D. Advantages of Sensitivity Based Link Addition

The sensitivity based link insertion algorithm has the following merits when compared to the other existing algorithms:

- Because of the use of delay sensitivity, this method identifies the sink pairs that are most prone to the effects of variations.
- Links insertion can be modified based on a given variation model. For example, if the power-supply noise model is available, the works of [1–5] cannot use that information while choosing the links. However, since our scheme directly uses the variation information in the form of delay sensitivity, the links selected will be more effective.
- The work of [6], though can make use of given variation information, is extremely slow. This is mainly because [6] updates the values of mean and variance for all sink delays after every single link addition while considering all

possible variation parameters simultaneously. However, since our method of obtaining the sensitivity information is a one-time process, our overall approach is much faster than the approach of [6].

- The calculation of delay sensitivity is compatible with higher order models including SPICE. Hence our algorithm can be applied even when high accuracy is required. In the case of using SPICE, we can employ a method very similar to the one used in [15] to obtain the delay sensitivities.
- Our algorithm does not involve the use of any empirical parameters such as α, β, γ , etc., as in [1–4].

In the next section, we present the experimental results comparing our sensitivity based methods with existing methods.

4. EXPERIMENTAL RESULT

In order to verify the effectiveness of the sensitivity based link insertion, we should ideally compare our results with that of the works [1–6]. Since the skew reduction in [6] was lesser than the works of [1–5] and also the benchmarks used were smaller, it is sufficient to compare our results with the best from the works of [1–5]. However, the work of [3] uses special tunable buffers and the work does not give details about the buffer. The works of [1, 4, 5] mainly focus on unbuffered clock trees as opposed to the focus on buffered clock trees in this work. Because of these reasons, we compare our results only with the work of [2].

Similar to [2], we use the benchmarks r1-r5 downloaded from GSRC Bookshelf [17]. We run SPICE based Monte Carlo simulations with 500 trials with interconnect width as a source of variation. We assume that the interconnect widths are Gaussian random variables with a σ value of 5%. Please note that we use this process variation model only as an example. Any other variation can also be used in our method. We use the 180nm technology parameters for the buffers and interconnect parameters. We implemented our algorithm in C++ and all the experiments were run on a 3.25GHz, 2GB Linux system. Table I shows the details of the buffered clock trees used in our experiments. The clock trees were obtained using the algorithm of [18], which was also implemented using C++. The column denoted by WL gives the wire-length of the trees. The worst case skew and the standard deviation values for each of the clock trees is given under the columns of WCS and SD respectively.

TABLE I: SIZE AND SKEW VARIABILITY INFORMATION FOR BUFFERED CLOCK TREES. WCS AND SD IN PICO-SECONDS

TC	# of Sinks	# of Buf	WL	WCS	SD
r1	267	32	1632666	183.4	86.2
r2	598	66	3257889	343.0	69.2
r3	862	85	3949749	293.0	63.0
r4	1903	184	8243951	360.7	88.5
r5	3101	281	12281280	361.0	91.6

Table II presents the skew variability information for the link-based non-trees obtained using the sensitivity based link

insertion method. Please note that all the values given in Table II are in terms of the values of the trees shown in Table I.

TABLE II: SKEW VARIATION INFORMATION FOR LINK BASED NON-TREES W.R.T. THE RESULTS OF TREES SHOWN IN TABLE I

TC	WL	SD	WCS	CPU(s)
r1	1.07	0.55	0.61	0.04
r2	1.03	0.76	0.86	0.12
r3	1.10	0.59	0.70	0.18
r4	1.11	0.60	0.82	0.74
r5	1.07	0.64	0.74	2.23
Average	1.076	0.62	0.74	0.662
% Change	+7.6	-38.0	-26.0	-

From Table II, we can see that the sensitivity based link insertion has achieved a 26% reduction in worst case skew, 38% reduction in standard deviation at the cost of 7.6% increase in the total wire length when compared with the buffered clock trees. In comparison, the method of [2] has achieved a 22% reduction in worst case skew, 30% reduction in standard deviation at the cost of 15% increase in wire-length. Thus, the sensitivity based method has achieved higher skew variability reduction while reducing the total wire-length increase by 7% compared to [2]. This indicates that the sensitivity based method is able to insert links much more efficiently than the method of [2].

Another important fact to be noted in Table II is that our sensitivity based algorithm is orders of magnitude faster when compared with the work of [6]. The work of [6] is a good comparison for run-time because it is the only work among [1–6] which can potentially make use of any given process variation model. Even for our biggest benchmark, the run time is in a few seconds when compared to several hours of run-time for the method of [6]. Thus, our sensitivity based algorithm is very fast and achieves better skew variability reduction using lesser routing resources.

5. CONCLUSION AND FUTURE WORK

In this paper, we have presented a fast and efficient sensitivity based link insertion algorithm. Unlike existing algorithms, our method is compatible with the use of higher order delay models and can efficiently use any given variation model. Experimental results show that our method is able to achieve better skew variability reduction compared with existing methods while utilizing lesser routing resources. In future, this work might be extended by incrementally adding links based on the sensitivity. The main challenge in obtaining such a method is to be able to efficiently update the sensitivity values after each link addition. This problem is non-trivial because the addition of the first link converts the tree into a non-tree, which complicates the analysis. We plan to investigate this topic in our future research.

6. ACKNOWLEDGEMENTS

This work is partially supported by SRC under contract 2005-TJ-1321 and IBM Faculty Award.

REFERENCES

- [1] A. Rajaram, J. Hu, and R. Mahapatra, "Reducing Clock Skew Variability via Cross Links," in *Proc. of IEEE/ACM DAC*, San Francisco, CA, pp. 55-62, Apr. 2004.
- [2] A. Rajaram and D. Z. Pan, "Variation tolerant buffered clock network synthesis with cross links," in *Proc. of ISPD*, San Francisco, CA, April 2006
- [3] G. Venkataraman, N. Jayakumar, S. Khatri, A. Rajaram, P. McGuinness, and C. Alpert, "Practical techniques to reduce skew and its variations in buffered clock networks," in *Proc. of IEEE/ACM ICCAD*, pp. 592-596, 2005.
- [4] A. Rajaram, D. Z. Pan, and J. Hu, "Improved Algorithms for Link Based Non-tree Clock Network for Skew Variability Reduction," in *Proc. of ISPD*, San Francisco, CA, April 2005.
- [5] A. Rajaram and D. Z. Pan, "Fast Incremental Link Insertion in Clock Networks for Skew Variability Reduction," in *Proc. of ISQED*, San Jose, CA, Mar. 2006
- [6] W.-C. D. Lam, J. Jain, C.-K. Koh, V. Balakrishnan, and Yiran Chen, "Statistical based link insertion for robust clock network design," in *Proc. of IEEE/ACM ICCAD*, pp. 588-891, Nov. 2005.
- [7] S. Lin and C. K. Wong, "Process-variation-tolerant Clock Skew Minimization," in *Proc. IEEE/ACM ICCAD*, pp. 284-288, Nov. 1994.
- [8] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani, "Impact of process variations on clock skew," in *Proc. of ISQED*, pp. 129-132, Mar. 2002
- [9] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas, "Impact of interconnect variations on the clock skew of a gigahertz micro-processor," in *Proc. of DAC*, pages 168-171, 2000.
- [10] M. Cho, S. Ahmed, and D. Z. Pan, "TACO: temperature aware clock-tree optimization," in *Proc. of the IEEE/ACM ICCAD*, San Jose, CA, Nov. 2005.
- [11] R. Saleh, S. Z. Hussain, S. Rochel, and D. Overhauser "Clock skew verification in the presence of IR-drop in the power distribution network," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. Vol. 19, No. 6, June 2000
- [12] R.-S. Tsay, "Exact zero skew," in *Proc. of the ICCAD*, Santa Clara, CA, pages 336-339, November 1991.
- [13] Z. Li, X. Lu, and W. Shi, "An algorithm for process variation reduction based on SVD," in *Proc. of IEEE ISCAS*, Bangkok, Thailand, Vol. 4, pp. 672-675, May 2003.
- [14] A. Gattiker, S. Nassif, R. Dinakar, and C. Long "Timing yield estimation from static timing analysis," in *Proc. of ISQED*, Mar.'01, pp. 79-84.
- [15] K. Wang and M. Marek-Sadowska, "Buffer sizing for clock power minimization subject to general skew constraints," in *Proc. of IEEE/ACM DAC*, pp. 159-164, 2004.
- [16] P. K. Chan and K. Karplus, "Computing signal delay in general RC networks by tree/link partitioning," in *Proc. of DAC*, pp. 485-490, Jun. 1989.
- [17] <http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST/>
- [18] Y. P. Chen and D. F. Wong, "An Algorithm for Zero-skew Clock Tree Routing with Buffer Insertion," in *Proc. of the EDTC*, Paris, France, pp. 230-236, Mar. 1996.