ANALYSIS AND OPTIMIZATION OF VLSI CLOCK DISTRIBUTION NETWORKS FOR SKEW VARIABILITY REDUCTION

A Thesis

by

ANAND KUMAR RAJARAM

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

August 2004

Major Subject: Electrical Engineering

ANALYSIS AND OPTIMIZATION OF VLSI CLOCK DISTRIBUTION NETWORKS FOR SKEW VARIABILITY REDUCTION

A Thesis

by

ANAND KUMAR RAJARAM

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

J. Hu (Co-Chair of Committee)	R. N. Mahapatra (Co-Chair of Committee)
D. M. H. Walker (Member)	W. Shi (Member)
J. Silva-Martinez (Member)	C. Singh (Head of Department)

August 2004

Major Subject: Electrical Engineering

ABSTRACT

Analysis and Optimization of VLSI Clock Distribution

Networks for Skew Variability Reduction. (August 2004)

Anand Kumar Rajaram, B.E, Anna University

Co–Chairs of Advisory Committee: Dr. Jiang Hu
Dr. Rabi Mahapatra

As VLSI technology moves into the Ultra-Deep Sub-Micron (UDSM) era, manufacturing variations, power supply noise and temperature variations greatly affect the performance and yield of VLSI circuits. Clock Distribution Network (CDN), which is one of the biggest and most important nets in any synchronous VLSI chip, is especially sensitive to these variations. To address this problem variability-aware analysis and optimization techniques for VLSI circuits are needed. In the first part of this thesis an analytical bound for the unwanted skew due to interconnect variation is established. Experimental results show that this bound is safer, tighter and computationally faster than existing approaches. This bound could be used in variation-aware clock tree synthesis.The second part of the thesis deals with optimizing a given clock tree to minimize the unwanted skew variations. Non-tree CDNs have been recognized as a promising approach to overcome the variation problem. We propose a novel non-tree CDN obtained by adding cross links in an existing clock tree. We analyze the effect of the link insertion on clock skew variability and propose link insertion schemes. The non-tree CDNs so obtained are shown to be highly tolerant to skew variability with very little increase in total wire-length. This can be used in applications such as ASIC design where a significant increase in the total wire-length is unacceptable.

To

My Parents

ACKNOWLEDGMENTS

First of all, I thank my parents for their blessings for my educational ambitions. It is because of their blessings and sacrifices that I have been able to reach my current state.

I thank my advisors, Dr. Jiang Hu and Dr. Rabi Mahapatra, for guiding me in my research for the last two years. Without their guidance and assistance I could not have conducted my research. I express my deep gratitude to both of them for their continued support.

I would like to thank Dr. Hank Walker and Dr. Weiping Shi for their advice on technical and career related matters. I also thank all of my committee members for taking so much time to help me with this thesis.

My past two years at Texas A&M have been a very good learning experience for me - both academic and otherwise. My stay here would not have been enjoyable but for my office mates. I wish to thank Praveen, Junyi, Siddharth, Purna, Nitesh and Subratha for making my life at A&M memorable. I thank Praveen and Junyi for their advise in both technical and non-technical things. I also enjoyed my several discussions with my office mates on a whole range of topics from realization of God to Microsoft vs Linux!

TABLE OF CONTENTS

CHAPTER Page

I INTRODUCTION 1
A. Analytical Bound for Clock Skew 2
B. Cross-link Based Non-tree CDN 2

II BACKGROUND AND PROPOSED RESEARCH 4
A. Reliable Worst Case Clock Skew Estimation 4
B. Clock Skew Variation Reduction and Non-tree Clock Routing . 5

III ANALYTICAL BOUND FOR UNWANTED SKEW DUE TO INTERCONNECT VARIATION 9
A. Introduction 9
B. Key Observation 10
C. Problem Analysis 11
D. The Minimum Delay Wire Shaping for Path with Branch Load . 14
E. The Maximum Delay Wire Shaping 16
F. The Skew Bound Depends on the Common Path 19
G. Experimental Results 22

IV REDUCING SKEW VARIABILITY BY CROSS LINK ADDITION 26
A. Introduction 26
B. Skew in RC Network 27
1. Delay In RC Network 27
2. Skew Variability Between Link Endpoints 28
3. Skew Variability Between Any Equal Delay Nodes 31
C. Link Insertion Based Non-tree Clock Routing 33
1. Algorithm Overview 33
2. Selecting Node Pairs for Link Insertion 34
a. Rule Based Selection Scheme 34
b. Minimum Weight Matching Based Selection 35
3. Non-zero Skew Routing 38
D. Experimental Results 38

V CONCLUSION 42

Page

REFERENCES 43

VITA 48

LIST OF TABLES

TABLE		Page
I	Comparison of computation time in seconds.	24
II	Maximum skew variation (MSV), standard deviation (SD) and total wire-length of trees. The CPU time is from running BST code.	39
III	Skew variations and wire-length in terms of tree results. Size of a tree+link network is the number of links. Size of a mesh is $\#rows \times \#columns$.	40

LIST OF FIGURES

FIGURE		Page
1	Non-tree clock networks: (a) center chunk; (b) spine; (c) leaf level mesh; (d) top level mesh. Each sink corresponds to a register or a clock subnetwork.	7
2	When estimating the worst case skew between sink s_1 and s_4 in (a), the clock tree can be reduced to a simpler model in (b).	12
3	The worst case skew estimation is equivalent to finding wire shaping to maximize/minimize the delay.	13
4	The branch load function.	14
5	Case 6 for the minimum delay wire shaping, the wire width is w_{lo} in $[0,g]$ and w_{hi} in $[h,L]$. Between g and h, the wire shape follows exponential function.	16
6	The delay function w.r.t. wire width is a convex function. The maximum/minimum delay wire width depends on the overlap between this function and wire width range $[w_{lo}, w_{hi}]$.	17
7	The maximum delay wire shaping.	18
8	Histogram of difference compared with Monte Carlo simulation for the maximum skew.	23
9	Histogram of difference compared with Monte Carlo simulation for the minimum skew.	23
10	Cross link insertion.	27
11	Skew variation vs. link position from both SPICE and Elmore delay model.	30
12	Tune location x of tapping point p such that nominal skew between sinks in sub-tree T_i and T_j is same as specifications. If there is great imbalance between T_i and T_j, wire snaking may be necessary as in (b).	33

FIGURE Page

13 Main algorithm of selecting node pairs for link insertion. 36

14 A bipartite graph model for selecting 4 node pairs between two sub-trees. Each node corresponds to a sub-sub-tree in a sub-tree. An edge weight is the shortest Manhattan distance between leaf(sink) nodes of two sub-sub-trees. 37

CHAPTER I

INTRODUCTION

When the VLSI feature size becomes progressively smaller, moving into the Ultra-Deep Sub-Micron Technology era, previously negligible variation effects start to affect circuit performance and yield significantly. Some of the most important variation effects are manufacturing variations [1], temperature variation and power supply noise [2], which are very difficult to be modeled. In many cases, the information about the variation effects are not available during design time, making it even more difficult to control them.

In synchronous VLSI circuits, the Clock Distribution Networks (CDNs) are especially affected by these variation effects. Since the complete synchronous circuit operation is being coordinated by the clock signal, any unwanted skew in the CDNs will adversely affect the performance of the entire chip. Also, since the clock skew is a lower bound for clock period, the unwanted skew due to variations form a bottleneck preventing improvement on clock frequency and thereby circuit operational speed.

In order to overcome the variation problem and to make circuits work reliably even in the presence of variations, a *variation-aware* circuit design methodology is needed. Two important aspects of any variation-aware circuit design methodology are 'variation modeling & analysis' and 'variability-aware optimization'. In variation modeling & analysis, the objective is usually to derive a model for a particular variation effect that can be used to predict the circuit performance. For example, one can obtain a model for effect of temperature variation on circuit performance. Once such a model has been obtained, that can be used *during* the planning and design phase of the circuit to make sure that the circuit will operate under all temperature conditions. In variability-aware optimization, the objective

---

The journal model is *IEEE Transactions on Automatic Control.*

is to come-up with novel optimization methods which make the circuits more tolerant to variations. One such example is the wire-sizing optimization [3] done so as to make the clock tree more tolerant to wire-width variation.

A. Analytical Bound for Clock Skew

In the first part of this thesis, we derive an analytical bound for the unwanted skew cause by the variations in the interconnect width. Interconnect variations result from many non-ideal conditions in manufacturing process such as etching error, mask mis-alignment and spot defects. For clock networks, these variations cause clock skew variations or unwanted skews. In modern high performance circuit designs, process variation induced clock skew is taking a greater and greater portion of clock period time. It is found in [4] that interconnect variations may cause up to 25% clock skew variations. Therefore, it is very important to model the impact of interconnect variations on clock network performance so that it can be considered during circuit/clock design.

In this work, we concentrate on the effect of wire width variation on clock skew. Interconnect width is the dominating factor compared to other wire parameters such as wire thickness because wire width shrinks faster under the non-uniform technology scaling and is considerably smaller than the thickness. We attempt to find an analytical bound for unwanted clock skew due to wire width variation using wire-shaping analysis used in delay driven Physical Design optimizations. Experimental results shows that our skew bound is safer and faster than existing methods.

B. Cross-link Based Non-tree CDN

In the second part of the thesis, we attempt to address the problem of making a CDN tolerant to the variation effects. There are several ways in which a CDN can be made to be tol-

erant to the variations. Some of them are buffer sizing and interconnect width sizing[5, 6], process variation aware routing[7, 8] and non-tree clock routing[9, 10]. Among these methods, the non-tree routing is usually the most effective because of the existance of redundant paths. Existence of redundant paths makes the delays in the sinks highly correlated and that can be used to make the skew variation very small.

In this work, we propose a new type of non-tree clock routing which is obtained by inserting links in an existing clock tree. We analyze the effects of inserting links in a clock tree using the tree/link delay evaluation work[11]. Based on the analysis, we propose link insertion algorithms which will convert a given clock tree to non-tree quickly. The links are inserted in the locations where they will be most effective to reduce skew variability. Experimental results on the link based non-tree CDNs so obtained shows that they are significantly tolerant to the skew variability with very little increase in the wire-length and power.

CHAPTER II

BACKGROUND AND PROPOSED RESEARCH

This chapter talks about some of the previous works that has been done in the areas of process variation modeling & analysis, clock skew estimation and reduction. It also introduces the basic idea of our research in both analysis and optimization for skew variability.

A. Reliable Worst Case Clock Skew Estimation

Realizing the importance of process variations, many works have been carried out to model their effect [12, 13, 14, 15, 16, 17, 18, 19]. One of the important objectives of these modeling works is to find a reliable estimation on the worst case timing performance induced by process variations. This can be either the worst delay along timing critical paths in timing analysis or the worst skew in a clock network. The estimation results can be applied as a feedback to guide further design iterations.

One common approach to find the worst case performance is to run Monte Carlo simulations for a certain number of iterations and pick the worst case performance among the results. In order to obtain a reliable estimation, the number of iterations is generally very large and consequently the high computational cost makes this method impractical for use during design. Another straightforward technique is to estimate the performance only at corner points of process variations. Even though this technique is computationally fast, it is overly pessimistic in estimating the worst case delay due to gate process variations as the correlations among the gates are neglected. Therefore, the corner-point technique is useful only when a loose bound needs to be found quickly. In seeking a good balance between estimation quality and runtime, probability based approaches [13, 19] and interval analysis method [12] has been proposed to establish a bound for the worst case timing performance.

Our proposed technique is based on the observation that estimating the worst case skew due to wire width variation is closely related to the non-uniform wire sizing problem in physical optimizations. That is, estimating the worst case skew between any two sinks can be viewed as the problem of finding min/max delay wire shaping. Using this key observation, we derive an analytical bound for the worst case skew between any two sinks. Experimental results show that our bound is both safer and faster than interval analysis method [12].

B. Clock Skew Variation Reduction and Non-tree Clock Routing

Aimed to reduce the effect of variation on clock skew, numerous clock routing works have been proposed [5, 6, 8, 9, 20, 21]. Among these works, non-tree clock routing [6, 8, 9, 20, 21] is a promising approach, since clock signal that propagates through multiple paths can compensate each other on variations.

Existing non-tree clock networks can be categorized as 1-dimensional structure [8, 21] and 2-dimensional structure [6, 9, 20]. One of the first non-tree clock routing works[8] is a 1-dimensional approach (see Figure 1(a)). In the method of [8], a fat metal piece, which is called center chunk, is placed in each subregion and the sinks in each subregion is connected to it directly. A center chunk is driven by a binary tree from the clock source. Since a center chunk is fat and driven at multiple points, the skew between any points on it is negligible.

Another similar approach is the spine method[21] which is employed in $Pentium^{TM}4$ microprocessor and illustrated in Figure 1(b). Obviously, a spine in [21] plays a role similar to the center chunk in [8]. One limitation of the 1-dimensional structures is that the variation of skew between different regions are not handled.

The 2-dimensional non-tree structure is also called mesh which can be at either leaf

level (close to clock sinks) or top level (close to the clock source). In the leaf level mesh approach [6, 20](Figure 1(c)), a metal wire mesh is overlaid on the entire chip area and driven at multiple points directly from clock source [6] or through a routing tree [20]. Each clock sink is connected to the nearest point on the mesh. This technique is proved to be very effective on suppressing skew variations in microprocessor designs. A leaf level mesh normally consumes enormous wire resources and power. Moreover, it is hard to be integrated with clock gating, which is a major power reduction technique. Therefore, its application is mainly restricted to high-end products such as microprocessors. In contrast, a top level mesh(Figure 1(d)) may consume less wire and power. In the top level mesh approach [9], the clock source drives a coarse mesh directly and clock sub-trees are attached to the mesh. The skew variations on the mesh are negligible, but skew variations within each sub-tree still exist and they may not be negligible.

Even though several non-tree schemes have been suggested and applied in realistic designs, there are still some important questions without clear and thorough answers.

- Why does a non-tree network have lower variability compared with a tree? Existing answers are based on either common sense or empirical simulation results. However, no theoretical or rigorous explanation has been provided yet.
- Does a non-tree clock network have to be a regular structure? If this restriction is relaxed, a huge solution space of non-tree topology can be explored. The poor tractability of timing performance in an arbitrary non-tree needs to solved before it is utilized.
- What is the most efficient usage of wire resource for reducing skew variability? Existing non-tree methods often consume excessive amount of wire-length and power. For example, the top level meshes in [9] result in 59% – 168% more wire area than

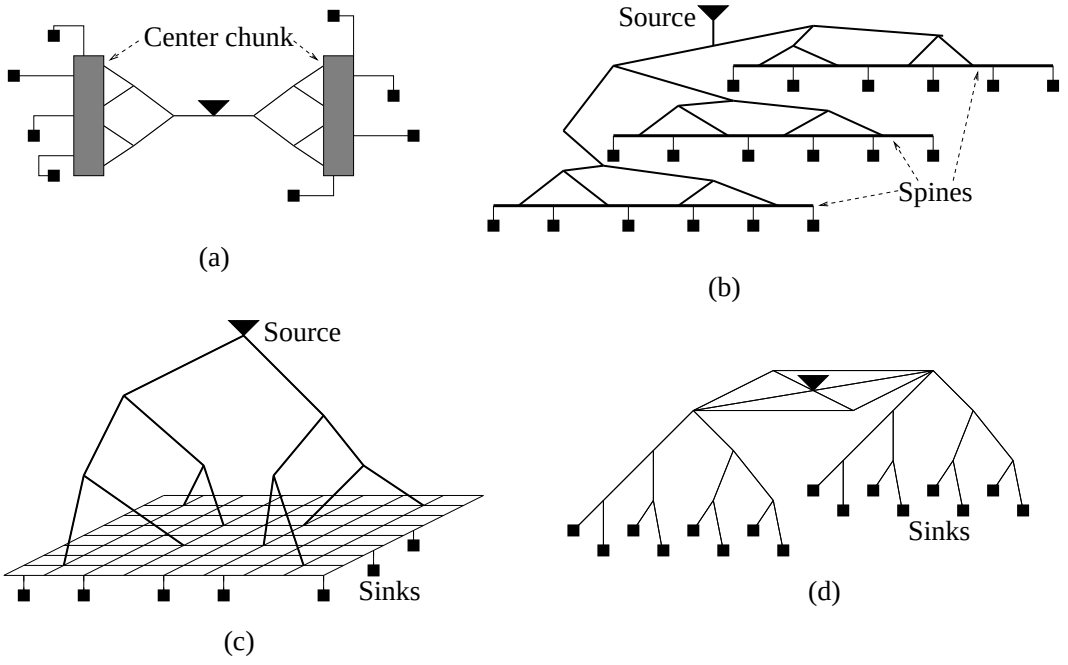

Fig. 1. Non-tree clock networks: (a) center chunk; (b) spine; (c) leaf level mesh; (d) top level mesh. Each sink corresponds to a register or a clock subnetwork.

trees. High expense on wire and power is rarely acceptable in ordinary chip designs except in the case of a few high-end products. Therefore, low cost non-tree networks are strongly needed, particularly for ASIC designs.

- Can non-tree topology be applied to achieve useful non-zero skew efficiently? Useful non-zero skew routing[5, 22, 23, 24] becomes more important for the sake of timing[25] and power/ground noise reduction[26].

In this work, efforts are made toward solving the above problems. We propose another framework of non-tree network which is constructed by inserting cross links in a clock tree. An analysis on impact of link insertion on skew variability is performed. The result of this analysis partially explains the reason why a non-tree may work better than a tree on skew variation reduction. In this structure, cross links can be inserted where they are most effective, so that a great wire efficiency can be obtained. Moreover, links can be applied to achieve variation tolerant non-zero skew easily. We suggest link insertion schemes that can quickly convert a clock tree to a non-tree with significantly lower skew variability and very small increase of wire-length. This method provides a low cost alternative to the existing non-tree methods. Monte Carlo simulations on benchmark circuits show that our method can achieve remarkable skew variability reduction with very little increase of wire-length.

CHAPTER III

ANALYTICAL BOUND FOR UNWANTED SKEW DUE TO INTERCONNECT VARIATION

In this chapter, we describe our work on the analytical bound for the unwanted skew due to interconnect width variation. This chapter briefly introduces the problem that is being attempted and emphasizes some of the important points that has been mentioned in the previous two chapters. The key observation that is used in deriving the analytical bound is described in Section B. The problem nature is analyzed in Section C. The minimum delay wire shaping for path with branch loads is described in Section D. In Section E, the maximum delay wire shaping is derived. The dependence of skew variation bound on the common path is discussed in Section F. The experimental results are provided in Section G.

A. Introduction

The impact of interconnect variation on clock skew is intrinsically hard to be modeled efficiently. This is because the worst case interconnect delay does not always occur at the process variation corner points. This fact makes the corner-point technique not applicable for interconnect variations. Furthermore, interconnect variations is distributive in nature in contrast to the localized variations for transistors. Since an interconnect may span a long distance, using a single variable to model each process parameter such as wire width or thickness is inadequate to capture the different process variation levels in different regions. This is especially true when the intra-chip variations start to dominate the inter-chip variations[1]. A naive approach to solve this problem is to segment a long wire into smaller pieces, and consider the variation of each piece individually. However, this approach may increase the number of variables considerably and thereby slow down the estimation speed.

B. Key Observation

The key observation in deriving our analytical bound is : *the problem of obtaining the worst case skew in a clock tree due to wire-width variation is very similar to the problem of delay-driven optimization of wire width in Physical Design*. That is, the problem of obtaining the worst case skew between any two sinks can be broken down to the problem of finding the wire-shaping that gives the minimum and maximum delay from the source to the sink. Once the wire-shaping that gives max/min delays has been obtained, the worst case skew between any two sinks will be the difference between the maximum delay at one sink and the minimum delay at the other or vice versa. Also, since this results in closed form expressions, the evaluation of the skew bound is very quick.

The minimum delay non-uniform wire sizing problem for a 2-pin wire(single load path) has been solved in [27, 28]. We derive the maximum wire shaping function for both single load path and multi-pin trees. The minimum delay wire shaping formula for multi-pin trees is also obtained in our work. Previously, a more general version of this problem was solved through an iterative algorithm [29]. Similar to the works of [27, 28, 29], we employ Elmore delay model for our derivations. Besides the bound, we discovered that the bound for skew between two clock sinks depends on their common upstream path, even though the skew between them is independent of the common path. This dependence is analyzed and found to be monotone in nature. These results establish an analytical bound for the unwanted skews due to wire width variation. Since the analytical bound can be computed very quickly, it can be applied to *process variation aware* clock network design as well as chip level design planning.

C. Problem Analysis

Given a pair of sinks s_1 and s_2 in a clock routing tree, our objective is to find the worst case skew or the skew bound due to wire width variation. The skew between the two sinks can be expressed as $q_{12} = t_1 - t_2$ where t_1 and t_2 are the delay from clock driver to s_1 and s_2, respectively. When there is wire width variation for the paths from driver to s_1 and s_2, the delay t_1 and t_2 vary in certain ranges of $[t_{1,min}, t_{1,max}]$ and $[t_{2,min}, t_{2,max}]$, respectively. Evidently, the worst case skew occurs at $q_{12,max} = t_{1,max} - t_{2,min}$ or $q_{12,min} = t_{1,min} - t_{2,max}$.

In most of the previous works, sometimes the skew is defined as $\max(|q_{12,max}|, |q_{12,min}|)$ and sometimes the term skew means the maximum absolute value of skews among all sink pairs in a clock network. The latter definition of global skew is usually for traditional zero-skew clock network. For modern aggressive VLSI designs, useful skews [24] are applied more frequently, thus, we use the concept of local pair-wise skew instead of the single global skew. In handling process variations and other delay uncertainties, target skews are usually specified as a set of permissible ranges [30] instead of a set of single values. Since there might be skew violation on both the upper-bound side and the lower-bound side of a permissible range, we consider the maximum and the minimum skew separately. It can be seen that the worst case skew can be obtained by estimating the maximum and the minimum delays under process variations.

In order to estimate the maximum and the minimum delays due to wire width variation, we reduce the routing tree into a simplified model demonstrated in Figure 2. In Figure 2(a), we wish to estimate the skew bound between sink s_1 and s_4. Since the *common upstream path* $s_0 \rightsquigarrow v_7$ for s_1 and s_4 does not contribute to the skew between them, we lump its wire resistance together with the driver resistance into a virtual driving resistor R at the *nearest common ancestor node* v_7 for s_1 and s_4 in Figure 2(b). Note that the value of R does affect the skew bound value, even though it does not contribute to skew! This will be explained

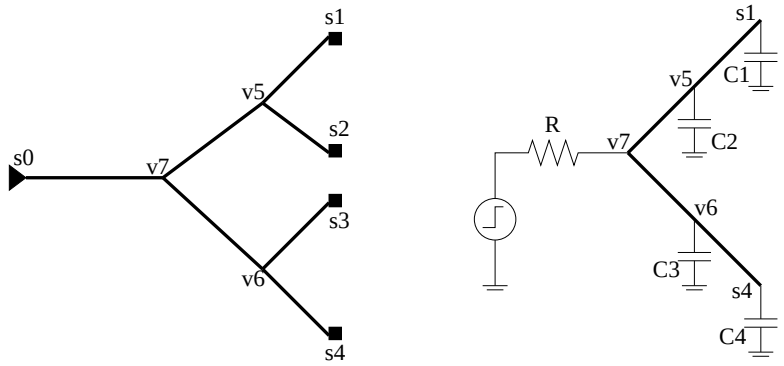

Fig. 2. When estimating the worst case skew between sink s_1 and s_4 in (a), the clock tree can be reduced to a simpler model in (b).

in detail in Section F. We call the branches $v_7 \rightsquigarrow s_1$ and $v_7 \rightsquigarrow s_4$ as *critical branches*. For wires off the critical branches such as $v_5 \rightsquigarrow s_2$ and $v_6 \rightsquigarrow s_3$ in Figure 2(a), their capacitance can be lumped to their load capacitance to get C_2 and C_3 at node v_5 and v_6, respectively. If we attempt to estimate the maximum(minimum) delay for sink s_1, the width of wire $v_5 \rightsquigarrow s_2$ should be the maximum(minimum) in order to maximize(minimize) the load C_2 for branch $v_7 \rightsquigarrow s_1$.

After the transformation in Figure 2, the worst case skew estimation between two sinks is reduced to estimating the maximum and the minimum delay of a path as in Figure 3 considering wire width variations. When the wire width w varies in the range of $[w_{lo}, w_{hi}]$, we need to find a wire shaping function $w(x)$ such that the delay from the virtual driver R to the sink is maximized or minimized. The minimum delay wire shaping function for a path without branch loads is solved in [27, 28]. An iterative wire shaping algorithm is provided in [29] to minimize a weighted sum of sink delays in a routing tree. Even though this algorithm can guarantee the optimal wire shaping solution and can be adopted directly in our case, the convergence rate of this algorithm has not been proved. Since we wish to minimize the delay to only one particular sink instead of a weighted sum of sinks delays, we are able to derive an analytical formula of wire shaping in Section D. The formula for

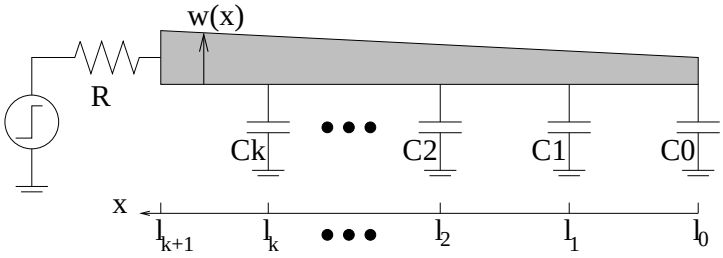

Fig. 3. The worst case skew estimation is equivalent to finding wire shaping to maximize/minimize the delay.

the maximum delay wire shaping is introduced in Section E.

The wire sheet resistance is denoted as r and the wire capacitance per unit length and unit width is represented as c. If there are k branch loads as indicated in Figure 3, we define the lumped downstream branch load capacitance as:

$$C_L(x) = \begin{cases} C_0 & : \quad 0 \le x < l_1 \\ C_0 + C_1 & : \quad l_1 \le x < l_2 \\ \vdots & : \quad \vdots \\ C_0 + C_1 + \ldots + C_k & : \quad l_k \le x \le l_{k+1} \end{cases} \tag{3.1}$$

This *branch load function* is depicted in Figure 4. Note that $l_0 = 0$ and l_{k+1} is same as the path length L. Then the downstream wire capacitance $C(x)$ at position x is $C_w(x) = c\int_0^x w(x)dx$. The total downstream capacitance can be written as $C(x) = C_L(x) + C_w(x)$ The Elmore delay of the path in Figure 3 can be expressed as:

$$t = RC(L) + r\int_0^L \frac{C(x)}{w(x)}dx \tag{3.2}$$

We can define the upstream resistance at position x as $R(x) = R + \int_x^L \frac{r}{w(x)}$. Fixing the wire shaping function $w(x)$ except an infinitesimal strip of width δ at z and let $w(z)$ to be a variable y. Then we may obtain the first order derivative of the delay function w.r.t wire

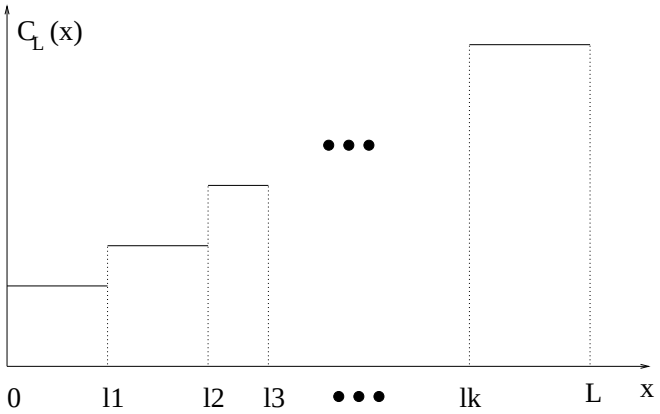

Fig. 4. The branch load function.

width as:

$$\frac{dt}{dy} = R(z - \frac{\delta}{2})c\delta - \frac{r\delta}{y^2}C(z + \frac{\delta}{2}) \tag{3.3}$$

Same conclusion is derived in [28] for the single load case, thus we omit the derivation here. From the above equation we can obtain $\frac{d^2t}{dy^2} = \frac{2r\delta}{y^3}C(z + \frac{\delta}{2}) > 0$. Thus the delay function is convex with respect to y or wire width.

D. The Minimum Delay Wire Shaping for Path with Branch Load

The minimum delay wire shaping for a path with single load $(k = 0)$ is shown in [27, 28]. In this section, we describe the minimum delay wire shaping for a path with multiple branch loads. Letting $q(x) = \frac{L}{2}\sqrt{\frac{rc}{RC_L(x)}}$, we first obtain the minimum delay wire shaping function when the wire width variation bound is not considered. (The proof is similar to [28])

Theorem 1*: The unconstrained minimum delay wire shaping function for a path with multiple branch loads is:*

$$w(x) = \frac{2C_L(x)}{cL}W(q(x))e^{2W(q(x))} \tag{3.4}$$

where $W(x) = \sum_{n=1}^{\infty}\frac{(-n)^{n-1}}{n!}x^n$ *is the Lambert's W function.*

For each wire segment between $l_i \leq x < l_{i+1} (0 \leq i \leq k)$, the wire shaping $w(x)$ is an exponential function. The overall wire shaping function is a piecewise exponential function which may be discontinuous at each branch point, since it depends on $C_L(x)$ which is a piecewise constant function whose value changes at the branch points. Even though there might be discontinuity, this wire shaping is monotonously increasing with respect to x [36].

When the bound on wire width variation $[w_{lo}, w_{hi}]$ is considered, the situation is more complicated. For wire segment between $l_i \leq x < l_{i+1} (0 \leq i \leq k)$, there are six cases that may occur:

- **Case 1:** The shaping of entire segment follows the exponential function as in Equation (3.4).
- **Case 2:** The width is uniformly w_{hi}.
- **Case 3:** The width is uniformly w_{lo}.
- **Case 4:** The width is w_{lo} when x is smaller than a value g, and the wire shaping follows exponential function from g to l_{i+1}.
- **Case 5:** The width is w_{hi} when x is greater than a value h, and the wire shaping follows exponential function from l_i to h.
- **Case 6:** The width is w_{lo} when x is smaller than a value g, w_{hi} when x is greater than a value h, and the wire shaping follows exponential function from g to h as shown in Figure 5

We call the position of $x = g$ and $x = h$ as *switching points*. The method to decide the switching points is very complicated as shown in [28]. Moreover, it is possible that all six cases need to be evaluated to find the exact minimum delay wire shaping. In practice, one can take the wire shaping according to Equation (3.4) without considering the wire width

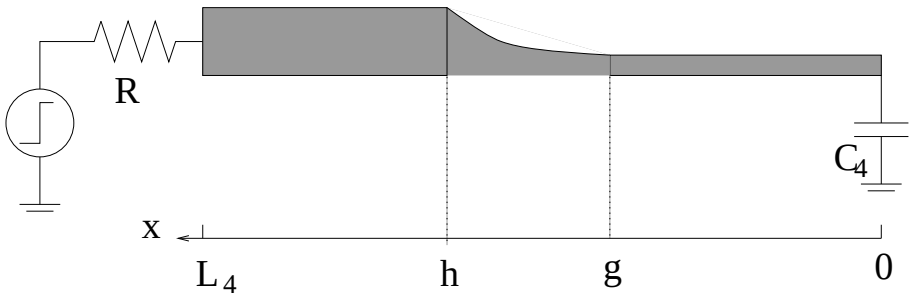

Fig. 5. Case 6 for the minimum delay wire shaping, the wire width is w_{lo} in $[0,g]$ and w_{hi} in $[h,L]$. Between g and h, the wire shape follows exponential function.

bound, and round the width to either w_{lo} or w_{hi} at where the width from Equation (3.4) exceeds the bound. The switching points can be found in the rounding process and the wire shaping in the exponential segment between $x = g$ and $x = h$ can be recomputed according to the updated upstream resistance and downstream capacitance. Note that this is slightly different from the rounding-alone heuristic mentioned in [28]. Even though this heuristic may result in suboptimal solutions, the computation becomes much easier and we observed that the error due to this is negligible in practice.

E. The Maximum Delay Wire Shaping

In this section, we will derive the maximum delay wire shaping for both single load path and path with multiple branch loads. For the ease of presentation, we start with the single load situation where $k = 0$. It has been shown in Section C that the delay is a convex function with respect to $w(x)$. This is illustrated in Figure 6. Therefore, $w(x)$ has to be either w_{lo} or w_{hi} to maximize the delay. Because of Equation (3.3), delay function with respect to $w(x)$ tends to be monotonously decreasing when x is large or the position is closer to the driver. Similarly, the delay function is more likely to be monotonously increasing when the position is closer to the sink side. This fact can be translated to the effect that $w(x)$ will be w_{lo} when x is large and $w(x)$ will be w_{hi} if x is small. When the value of x increases,

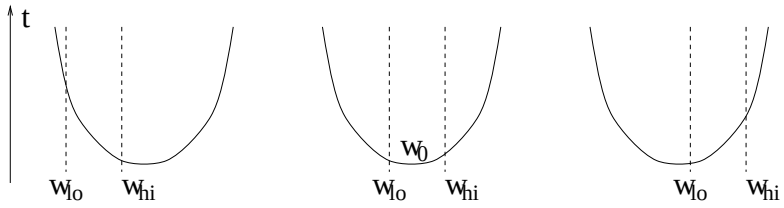

Fig. 6. The delay function w.r.t. wire width is a convex function. The maximum/minimum delay wire width depends on the overlap between this function and wire width range $[w_{lo}, w_{hi}]$.

the delay function with respect to wire width at x changes in the direction from (a) to (b) to (c) in Figure 6. Therefore, the maximum delay wire shaping looks like the example in Figure 7. The next problem is to find the partitioning point $x = p$ where the wire width is switched from w_{lo} to w_{hi}.

When $k = 0$, the Elmore delay for Figure 7 can be written as:

$$\begin{aligned} t &= \alpha p^2 + \beta p + \gamma \\ \alpha &= -\frac{rc(w_{hi} - w_{lo})}{w_{lo}} \\ \beta &= (w_{hi} - w_{lo})(Rc + \frac{rcL}{w_{lo}} - \frac{rC_0}{w_{lo}w_{hi}}) \\ \gamma &= RLcw_{lo} + RC_0 + \frac{1}{2}rcL^2 + \frac{rLC_0}{w_{lo}} \end{aligned} \tag{3.5}$$

In order to find the p that maximize the delay, we first find the derivative:

$$\frac{dt}{dp} = 2\alpha p + \beta \tag{3.6}$$

Since α is always negative, $\frac{d^2t}{dp^2} < 0$ and the maximum delay is reached when

$$p = \frac{1}{2}(\frac{Rw_{lo}}{r} + L - \frac{C_0}{cw_{hi}}) \tag{3.7}$$

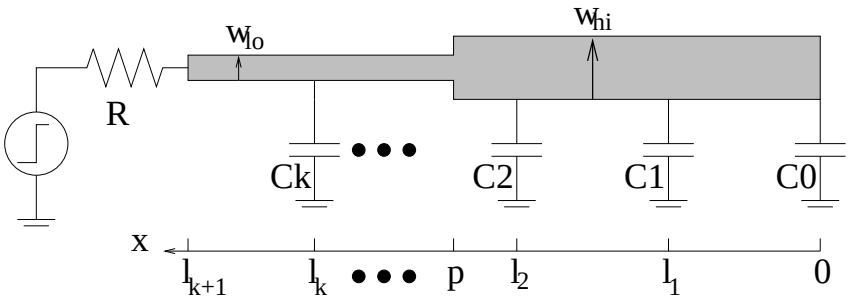

Fig. 7. The maximum delay wire shaping.

by letting $\frac{dt}{dp} = 0$. In other words, p satisfies the following equation:

$$\frac{R}{r/w_{lo}} + (L - p) = p + \frac{C_0}{cw_{hi}} \tag{3.8}$$

If we transform the driver into a piece of wire with width w_{lo} whose resistance is same as the driver resistance and treat the load C_0 as a wire of width w_{hi} with same value of capacitance C_0, then the above equation shows that p makes length of the fat segment the same as the length of the thin segment.

When we consider the situation with branch loads, i.e., $k > 0$, the properties on $\frac{dt}{dy}$ for Equation (3.3) do not change and the wire shape is same as in Figure 7 and the value of p is determined by:

$$\frac{R}{r/w_{lo}} + (L - p) = p + \frac{C_L(p)}{cw_{hi}} \tag{3.9}$$

Even though the wire shape in Figure 7 looks strange, it may happen in reality. If the path is routed in an L-shape with a horizontal and a vertical segment,the two segments are usually on different metal layers. Therefore, it is likely that the wire width has a abrupt change at layer switching.

F. The Skew Bound Depends on the Common Path

It is easy to see that the variation along the upstream common path for a pair of sinks does not contribute to the skew between them. For the example in Figure 2, the variation along path $s_0 \rightsquigarrow v_7$ has nothing to do with the skew value between sink s_1 and sink s_4. However, when the wire width at the common path changes, the resistance value R in Figure 2 also changes. Also, the maximum/minimum wire shaping from v_7 to s_1 and s_4 will change as well because both the min-delay wire shaping and max-delay wire shaping depends on the upstream path. Therefore, the skew bound is affected by the variation of R or their common path.

Theorem 2: *Given fixed switching points, the maximum(minimum) skew between two sinks is a convex(concave) function with respect to their common driving resistance.*

Proof: Let us consider the case where the maximum skew between two sinks s_1 and s_4 need to be computed. Let the distance from s_1 and s_4 to their nearest ancestor node v_7 be L_1 and L_4, respectively. The load capacitance at s_1 and s_4 are C_1 and C_4, respectively. The branch loads are temporarily ignored, the conclusion for path with branch loads is the same and can be extended from the single load case easily. The maximum skew between them can be expressed as $q_{14,max} = t_{1,max} - t_{4,min}$. From Section D and E, we know that both $t_{1,max}$ and $t_{4,min}$ depend on the value of R. We will analyze $\frac{dq_{14,max}}{dR}$ by evaluating $\frac{dt_{1,max}}{dR}$ and $\frac{dt_{4,min}}{dR}$.

By plugging Equation (3.7) and the expression of β into βp, we have $\beta p = -\alpha p^2$. Then the maximum delay Equation (3.5) can be transformed to:

$$t_{1,max}(R) = \alpha p^2(R) + \gamma(R)$$

Then we can obtain the derivative:

$$\frac{dt_{1,max}}{dR} = 2\alpha p(R)\frac{dp}{dR} + \frac{d\gamma}{dR} \tag{3.10}$$

$$= c(w_{hi} - w_{lo})p(R) + L_1 c w_{lo} + C_1$$

$$= \frac{cw_{lo}(w_{hi} - w_{lo})}{2r}R + \frac{1}{2}\left((w_{lo} + w_{hi})(L_1 c + \frac{C_1}{w_{hi}})\right) \tag{3.11}$$

The derivative of $t_{4,min}$ depends on the six different cases described in Section D. We will show the derivations for the most basic case 1 and the most complex case 6. Derivations for other cases are similar, and all these cases lead to the same conclusion.

For case 1, the equation for the minimum delay (for single load case) is given as [27]

$$t_{4,min} = \frac{1}{4}crL_4^2(1 + 2W(\hat{x}))/W(\hat{x})^2 \tag{3.12}$$

where $W(\hat{x})$ is the Lambert's W function and $\hat{x} = \frac{L_4}{2}\sqrt{\frac{cr}{C_4 R}}$. Since the Lambert's W function satisfies $W(\hat{x})e^{W(\hat{x})} = \hat{x}$, we can obtain its derivative as

$$W'(\hat{x}) = \frac{W(\hat{x})}{\hat{x}(1 + W(\hat{x}))} \tag{3.13}$$

Then we have the derivative of $t_{4,min}$ with respect to R:

$$\frac{dt_{4,min}}{dR} = \frac{rcL_4^2}{4W^2(\hat{x})R} \tag{3.14}$$

Combining Equation (3.10) and (3.14), we can get the second order derivative of $q_{14,max}$:

$$\frac{d^2 q_{14,max}}{dR^2} = \frac{cw_{lo}(w_{hi} - w_{lo})}{2r} + \frac{l_4^2 r^2 c}{4rR^2 W(\hat{x})(1 + W(\hat{x}))} \tag{3.15}$$

Evidently, $\frac{d^2 q_{14,max}}{dR^2} > 0$ and $q_{14,max}(R)$ is a convex function.

Now we consider case 6 which is more complex and general. The wire shaping for

case 6 is depicted in Figure 5. We can divide this wire into three segments: (i) the thin segment from $x = 0$ to $x = g$, (ii) the exponential segment between $x = g$ and $x = h$ and (iii) the fat segment from $x = h$ to $x = L_4$. We use $C_{thin} = cw_{lo}g$, $C_{exp} = \int_g^h cw(x)dx$ and $C_{fat} = cw_{hi}(L_4 - h)$ to represent the wire capacitance for each segment. Similarly, the wire resistance for each segment can be defined as $R_{thin} = \frac{rg}{w_{lo}}$, $R_{exp} = \int_g^h \frac{r}{w(x)}dx$ and $R_{fat} = \frac{r(L_4-h)}{w_{hi}}$.

We can find the wire delay for the exponential segment itself as:

$$\tilde{t} = \int_g^h \frac{r}{w(x)} \left(\int_g^x cw(z)dz \right) dx \tag{3.16}$$

For the exponential segment, we can treat the fat segment as part of its driving resistance and the thin segment as part of its load capacitance. Thus we can express the delay from $x = h$ to $x = g$ as:

$$t_{exp} = (R + R_{fat})(C_{exp} + C_{thin} + C_4) + \tilde{t} + R_{exp}(C_{thin} + C_4)$$

The value of t_{exp} can be obtained through Equation (3.12) except that the R is replaced by $\tilde{R} = R + \frac{r(L_4-h)}{w_{hi}}$ and the C_4 is replaced by $\tilde{C_4} = C_4 + gcw_{lo}$.

The total path delay from $x = L_4$ to $x = 0$ can be written as:

$$\begin{aligned}
t_{4,min} &= R(C_{fat} + C_{exp} + C_{thin} + C_4) \\
&\quad + R_{fat}(\frac{1}{2}C_{fat} + C_{exp} + C_{thin} + C_4) \\
&\quad + \tilde{t} + R_{exp}(C_{thin} + C_4) \\
&\quad + R_{thin}(\frac{1}{2}C_{thin} + C_4) \\
&= t_{exp} + RC_{fat} + \frac{1}{2}R_{fat}C_{fat} + R_{thin}(\frac{1}{2}C_{thin} + C_4) \\
&= \frac{1}{4}cr(h-g)^2(1 + 2W(\tilde{x}))/W(\tilde{x})^2 \\
&\quad + RC_{fat} + \frac{1}{2}R_{fat}C_{fat} + R_{thin}(\frac{1}{2}C_{thin} + C_4)
\end{aligned}$$

where $\tilde{x} = \frac{h-g}{2}\sqrt{\frac{rc}{\tilde{R}C_4}}$. Comparing the above equation and Equation (3.12), the differences are (i) there are additional terms with at most linear dependence on R and (ii) R is replaced by $\tilde{R}$ through a linear transformation. Since both differences have only linear dependence on R, they do not change the property of $\frac{d^2 t_{4,min}}{dR^2} > 0$ and $q_{14,max}(R)$ is still a convex function for case 6. Other cases can be proved in the same way.

For a path with multiple branch loads, the difference is that the constant load C_4 is replaced by the branch load function $C_L(x)$. Since the branch load function is piecewise constant, it does not change the property of $\frac{d^2 t_{4,min}}{dR^2} > 0$ either. Since $q_{14,min} = t_{1,min} - t_{4,max}$ is symmetric to $q_{14,max} = t_{1,max} - t_{4,min}$, we can conclude that $q_{14,min}$ is a concave function with respect to R. **Q.E.D.**

According to Theorem 2, we need to compute the wire shaping for $q_{14,max}$ twice, one with the minimum R by setting the wire width along the common path $s_0 \rightsquigarrow v_7$ to w_{hi}, the other with the maximum R by letting the wire width along the common path be w_{lo}. The maximum of the two skew results is finally selected as the worst case bound.

G. Experimental Results

In order to validate the bound we derived, we implemented our formulas, Monte Carlo and the interval analysis [12] for comparisons. Even though Monte Carlo method is not efficient, it can generate a reliable estimation on the worst case performance if the number of iterations is sufficiently large. Therefore, the result of Monte Carlo serves as an ideal baseline for comparisons. The reason to compare with interval analysis method is because its objective is very close to ours: to establish a bound efficiently.

The test cases are the $r1 - r5$ which are applied in the bounded skew clock routing (BST) work [35]. We downloaded the BST code from the GSRC bookshelf $(http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST/)$ and generated clock routing trees for

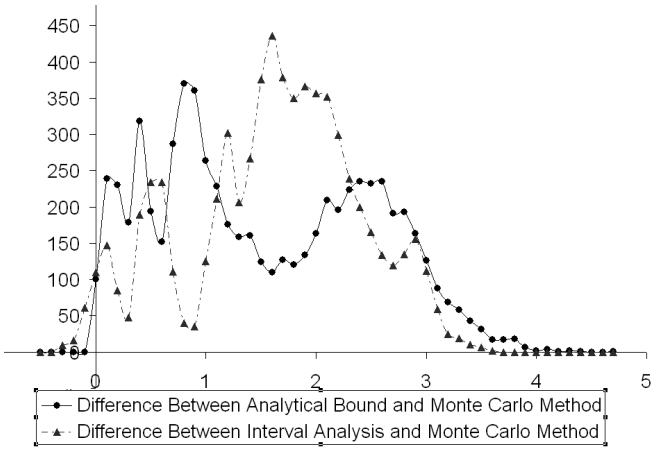

Fig. 8. Histogram of difference compared with Monte Carlo simulation for the maximum skew.

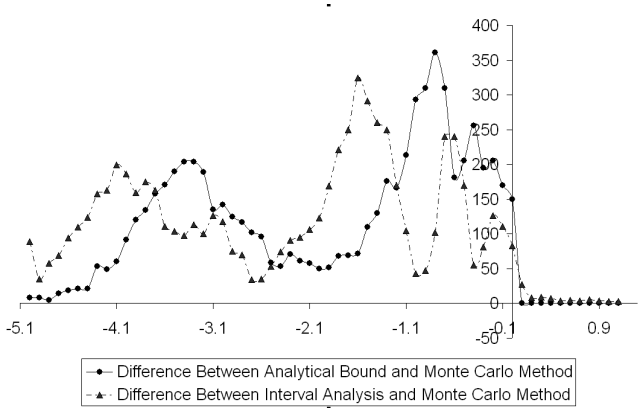

Fig. 9. Histogram of difference compared with Monte Carlo simulation for the minimum skew.

$r1-r5$ by running the BST code. The global skew bound is set to be $100ps$. We assume $\pm 30\%$ variations on wire width. We implemented our formulas, Monte Carlo and the interval analysis method in C language. The experiments are performed on a PC with Pentium III, 655 MHz processor and 512 MB memory.

We evaluated the skew bound due to wire width variations for 6725 pairs of sinks in the five testcases of $r1-r5$ by all of the three methods. In order to obtain a meaningful estimation, we segment long wires into small pieces of about $50\mu m$ for the Monte Carlo and interval analysis. Therefore, the wire width for each piece is an individual variable. For

each sink pair, we run Monte Carlo simulation for 50,000 trials when the width for each wire piece is selected randomly in the range of $[w_{lo}, w_{hi}]$. When estimating the minimum delay wire shaping we applied the heuristic described in Section D to decide the switching points and the optimal wire shaping for them. This heuristic brings great implementation convenience with negligible quality penalty.

Table I. Comparison of computation time in seconds.

Testcase	#sinks	#pairs	Monte Carlo	Interval	Ours
r1	267	266	8277	3	0.46
r2	598	597	35684	7	4.72
r3	862	861	70288	14	9.65
r4	1903	1902	180270	90	38.31
r5	3100	3099	408750	277	77.5

We take the result from Monte Carlo simulation as baseline. The result from our bound is evaluated by taking the difference between them. In other words, we evaluate the maximum/minimum skew from our bound minus the maximum/minimum skew from Monte Carlo simulation for each pair of sinks. The bound result from the interval analysis is evaluated in the same way. Figure 8 and 9 show the histograms of the difference for the maximum skew and the minimum skew, respectively. In Figure 8, the difference from our bound is almost alway greater than zero. Meanwhile, the difference from our bound for the minimum skew in Figure 9 is almost always non-positive. This fact tells that our method generally provides a bound for the worst case skew in practice, even though we applied heuristic in the implementation. Some sink pairs have similar path lengths to their nearest common ancestor node and the driver, thus they have similar skew variation behaviors. This fact results in serval peaks in the histograms. Figure 8 and 9 show that the peaks from our method are closer to zero difference compared to the peaks from the interval analysis. Thus, our method provides a tighter bound than the interval analysis. Moreover, our method

gives less number of under-estimations compared with the interval analysis.

The computation time for each method is shown in Table I. In Table I, column 2 and column 3 show the number of sinks and the number of sink pairs whose skew are evaluated. We can see that the runtime from the Monte Carlo simulation is impractical. Compared with interval analysis, the runtime of our method is always shorter.

The conclusion and future research for this chapter will be discussed in the chapter V of this thesis.

CHAPTER IV

REDUCING SKEW VARIABILITY BY CROSS LINK ADDITION

In this chapter, we describe our work on the cross link based non-tree topology. This chapter briefly introduces the problem that is being attempted and emphasizes some of the important points that has been mentioned in the introductory chapters. In Section B, we discuss about the skew in general RC networks. After our analysis of the skew in RC networks, we then describe about the effects and different scenarios of adding links in a clock tree. Based on this, we propose different link insertion schemes in Section C. Finally, in Section D we discuss about the experimental results for the different link insertion schemes.

A. Introduction

As one of the largest nets and one of the most frequently switching nets at the same time, the clock network has paramount influence on both energy efficiency and power/ground noise[26]. Therefore, the objective of clock network design has long been delivering zero clock skew[31] or useful non-zero skew[25] with a minimum size/wirelength[32, 24]. The unwanted skew variations in a CDN are not only harmful to timing performance but also difficult to control, because reliable estimations on these variations are generally not available during clock network design.

As discussed in chapter 2, one of the most effective methods in reducing clock skew variability is non-tree clock routing. Most of the existing non-tree routing belong to very simple and regular type of structures. If this requirement is relaxed, the resulting solution space is huge. The advantage of a large solution space is that it is likely that a good non-tree topology with less variability exists. But the obvious disadvantage is the problem of finding the good non-tree topology that satisfies our requirement in such big solution space. In this work, we attempt to address this problem by inserting cross links in a given clock tree.

B. Skew in RC Network

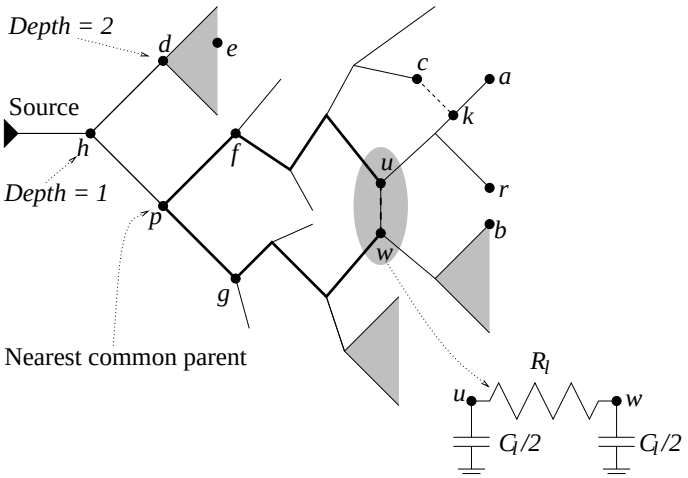

Fig. 10. Cross link insertion.

In this section, delay and skew variation of a non-tree RC network will be analyzed. Elmore delay model is employed due to its high fidelity[33] and ease of computation. A SPICE simulation is performed on a simple case to verify a conclusion from Elmore delay model.

1. Delay In RC Network

The Elmore delay at node i in an RC network is given by $t_i = \sum_j R_{i,j} C_j$ where C_j is the ground capacitance at node j. The transfer resistance $R_{i,j}$ is equal to the voltage at node i when $1A$ current is injected into node j and all other node capacitors are zero[10]. A non-tree RC network can be represented by a graph $G = (V, E)$ with the node set V composed by the source, sinks and Steiner nodes. The graph can be decomposed to a spanning tree $T = (V, E_T)$ and a set of link edges E_l such that $E = E_T \cup E_l$. As an alternative approach[11] more suitable for analysis, the delay from the source to each node can be evaluated by

starting with delays in the tree T and then incrementally inserting every link edge in E_l.

In Figure 10, a network is indicated by the solid lines and a cross link is inserted between node $u \in V$ and node $w \in V$. If the link has a wire resistance of R_l and wire capacitance of C_l, the link insertion can be decomposed to inserting a resistor of R_l between u and w and adding a capacitor of $\frac{C_l}{2}$ at node u and w. Adding link capacitors does not change the network topology,thus its effect can be estimated easily. If the Elmore delay from the source to any sink i is t_i before the link insertion, the delay $\tilde{t}_i$ after only adding the link capacitors is given by:

$$\tilde{t}_i = t_i + \frac{C_l}{2}(R_{i,u} + R_{i,w}) \tag{4.1}$$

The impact of the link resistance R_l can be analyzed through the technique by Chan and Karplus[11]. According to [11], the delay at node i is changed from $\tilde{t}_i$ to $\hat{t}_i$ given by:

$$\hat{t}_i = \tilde{t}_i - \frac{\tilde{t}_u - \tilde{t}_w}{R_l + r_u - r_w} r_i \tag{4.2}$$

where r_i,r_u and r_w are equal to the Elmore delay at i, u and w when $C_u = 1, C_w = -1$ and the other node capacitance are zero.

2. Skew Variability Between Link Endpoints

If a link is inserted between node u and node w, let us first look at its impact on skew between u and w. If the delay from the source to u and w are t_u and t_w, respectively, the skew between them is $q_{u,w} = t_u - t_w$. According to Equation(4.1) and Equation(4.2), the skew $\hat{q}_{u,w}$ after the link insertion is:

$$\hat{q}_{u,w} = \frac{R_l}{R_l + r_u - r_w}(q_{u,w} + \frac{C_l}{2}(R_{u,u} - R_{w,w})) \tag{4.3}$$

The effect of the link capacitance C_l and the link resistance R_l can be separated. The link capacitance often changes the skew value as the value of $R_{u,u}$ is often different from $R_{w,w}$.

The effect of the link resistance R_l can be found by neglecting C_l and the following equation.

$$\hat{q}_{u,w} = \frac{R_l}{R_l + r_u - r_w} q_{u,w} \tag{4.4}$$

Thus, the effect of R_l depends on the value of $q_{u,w}$. A case of our special interest is when the nominal skew between u and w is zero. In this case, R_l does not affect the nominal skew and $q_{u,w}$ may represent the unwanted skew due to variations. Then, we can reach the following useful conclusions.

Lemma 1: *If two distinctive nodes in an RC network have zero nominal skew between them, inserting a resistor between them always reduces their skew variability.*

Proof: When a resistor R_l is inserted between two distinctive nodes u and w which have zero nominal skew between them, then the skew variation between them, $q_{u,w}$, will be scaled by a factor of $\frac{R_l}{R_l + r_u - r_w}$ according to equation(4.4). Since r_u is obtained by computing the Elmore delay when $C_u = 1$, $C_w = -1$ and the capacitance at other nodes are zero, $r_u = R_{u,u} - R_{u,w}$. According to the computation of the transfer resistance, $R_{u,u} \geq R_{u,w}$ and $r_u \geq 0$. Similarly, $r_w = R_{u,w} - R_{w,w} \leq 0$. Since u and w are two distinct nodes with zero nominal skew, neither $r_u = 0$ nor $r_w = 0$ can happen. Thus, $r_u - r_w > 0$ is always true and therefore, $|\hat{q_{u,w}}| < |q_{u,w}|$ is always true.

Lemma 2: *In a clock tree, considering that a resistor R_l is inserted between two disjoint paths $p \rightsquigarrow b$ and $p \rightsquigarrow r$, where b and r are two leaf nodes and p is their nearest common ancestor node, and link endpoints $u \in p \rightsquigarrow r$ and $w \in p \rightsquigarrow b$ always have zero nominal skew, the variability of skew $q_{r,b}$ becomes smaller when the resistor is moved from p toward leaf nodes b and r.*

Proof: The resistor insertion is illustrated in figure 10. The variation of skew between r and b after inserting resistor R_l between u and w can be expressed as:

$$\hat{q_{r,b}} = \frac{R_l}{R_l + r_u - r_w} q_{u,w} + q_{ur,wb} \tag{4.5}$$

where $q_{ur,wb} = t_{u,r} - t_{w,b}$ is the difference between delays of path $u \rightsquigarrow r$ and path $w \rightsquigarrow b$. Since the RC network is a tree before inserting the resistance, r_u and $-r_w$ are the total resistance along path $p \rightsquigarrow u$ and $p \rightsquigarrow w$, respectively. Thus $R_{loop} = R_l + r_u - r_w$ is the total resistance along the loop of $p \rightsquigarrow u \rightsquigarrow w \rightsquigarrow p$. Original variation of skew between r and b is $q_{r,b} = q_{u,w} + q_{ur,wb}$. The effect of inserting the resistance is to scale the value of $q_{u,w}$ by $\frac{R_l}{R_{loop}} < 1$. When the resistor is moved from the nearest common ancestor towards r and b, $\frac{R_l}{R_{loop}}$ becomes smaller and smaller and greater portion of $q_{r,b}$ is scaled down.

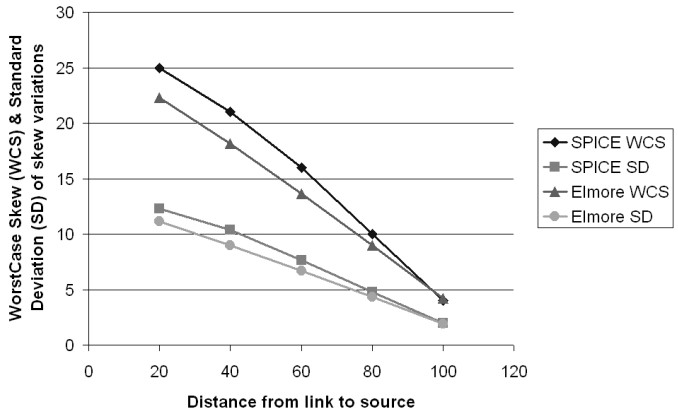

Fig. 11. Skew variation vs. link position from both SPICE and Elmore delay model.

Lemma 1 and Lemma 2 are based on Elmore delay model which is often criticized for its inaccuracy and particularly for neglecting the resistive shielding effect[34]. However, the resistive shielding effect is prominent only at nodes close to the source while clock sinks are generally far away from the source node. Hence, the inaccuracy of Elmore delay at clock sinks is usually insignificant. A SPICE simulation is performed on a simple case

to verify Lemma 2 and demonstrate the fidelity of Elmore delay model. In this simple case, two sinks are driven by a source directly. The two sinks have the same load capacitance and the same distance of $100\mu m$ from the source. We let the driver resistance, wire width and the sink capacitance have $\pm 15\%$ variation following a normal distribution. We obtain skew variation between the two sinks when a link is inserted between two source-sink paths at $20\mu m$, $40\mu m$, $60\mu m$, $80\mu m$ and $100\mu m$ away from the source. The size of the link is constant in each test. The worst case skew and the standard deviation of skew variation from both SPICE model and the Elmore delay model are plotted in Figure 11. This plot shows that there is strong correlation between SPICE results and Elmore delay results. In addition, the SPICE simulation results support the conclusion of Lemma 2.

3. Skew Variability Between Any Equal Delay Nodes

From Equation (4.1) and (4.2), the skew between two arbitrary node i and j after inserting link between u and w is:

$$\hat{q}_{i,j} = q_{i,j} + \frac{C_l}{2}(R_{i,u} + R_{i,w} - R_{j,u} - R_{j,w}) - \frac{\hat{q}_{u,w}}{R_l}(r_i - r_j) \tag{4.6}$$

Evidently, the link capacitance C_l usually changes the nominal skew. If only the link resistance R_l is considered, the skew becomes:

$$\hat{q}_{i,j} = q_{i,j} - \frac{r_i - r_j}{R_l + r_u - r_w} q_{u,w} \tag{4.7}$$

We consider the case where the nominal skew is zero between u and w as well as i and j. In this case, R_l does not affect the nominal skew between i and j. Further, both $q_{u,w}$ and $q_{i,j}$ can be treated as skew variations. Then, Equation(4.7) can be interpreted as that $q_{u,w}$ is scaled by $\frac{r_i - r_j}{R_l + r_u - r_w}$ and added to $q_{i,j}$. Whether the magnitude of $q_{i,j}$ is reduced depends on the signs of $q_{u,w}$, $q_{i,j}$ and $r_i - r_j$.

If we consider in the context of inserting links in a clock tree $T = (V, T_E)$, the node u

is in a sub-tree $T_f \subset T$ and the node w is in another sub-tree $T_g \subset T$ as shown in Figure 10. The root node of T_f and T_g are the two child nodes of the nearest common ancestor node p between u and w. The effect of R_l on $q_{i,j}$ depends on the locations of i and j in T and can be analyzed in three scenarios as follows.

Scenario 1: One of i and j is in sub-tree T_f and the other is in sub-tree T_g, ex., $i \in T_f$ and $j \in T_g$. Since i and u are in the same sub-tree, their delay t_i and t_u have certain correlation with respect to t_j or t_w. Similarly, delay t_j is more correlated with t_w than with t_i or t_u. Therefore, there is certain correlation between $q_{i,j}$ and $q_{u,w}$. If the correlation is sufficiently strong, $q_{i,j}$ and $q_{u,w}$ usually have the same sign. As $r_u - r_w \geq r_i - r_j \geq 0$ (proof is similar to Lemma 1), the link resistance may reduce the variability of skew between i and j. In a special case, when i is u and j is w, i.e., $q_{i,j}$ and $q_{u,w}$ have perfect correlation, the link resistance always reduces skew variability as stated in Lemma 1.

Scenario 2: Both i and j are in the same sub-tree T_f or T_g. In this scenario, r_i and r_j have same sign and $\frac{r_i - r_j}{R_l + r_u - r_w}$ is generally smaller than 0.5. Thus, the skew variation between i and j is increased or decreased by a small fraction of $|q_{u,w}|$. Since i and j are in the same sub-tree, their skew variation in original tree is usually not large either.

Scenario 3: One of i and j is in the sub-tree T_p and the other node is disjoint with T_p. For example, i is in T_p like b and j is not in T_p like d in Figure 10. If node j is disjoint with T_p, there is no overlap between any source-to-j path and T_p. Hence, $r_j = 0$ and there is no predictable correlation between $q_{i,j}$ and $q_{u,w}$. In the worst case of signs, $q_{i,j}$ has sign opposite to $q_{u,w} r_i$. Since the nominal skew between u and w is zero, $r_u \approx -r_w \approx |r_i|$ in general. Therefore, R_l may increase or decrease the skew variation between i and j by a half of the skew variation between u and w.

In summary, the link resistance R_l usually reduces skew variability in Scenario 1, may increase skew variability a little in Scenario 2, and may increase skew variability more in Scenario 3.

C. Link Insertion Based Non-tree Clock Routing

1. Algorithm Overview

Our objective is to design a clock routing algorithm that can achieve low skew variability and low wire consumption. Based on the analysis in Section B, we propose to construct the clock network by inserting cross links to a clock tree. In this incremental approach, many existing clock tree routing algorithms[32, 24] can be utilized and the non-tree clock routing problem becomes more manageable.

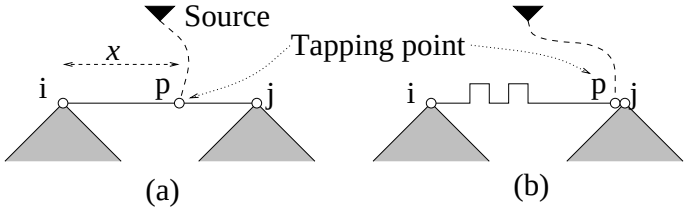

Fig. 12. Tune location x of tapping point p such that nominal skew between sinks in sub-tree T_i and T_j is same as specifications. If there is great imbalance between T_i and T_j, wire snaking may be necessary as in (b).

Each link insertion can be decomposed to adding link capacitance to its endpoints and inserting link resistance. Based on the analysis in Section B, the nominal skews may be affected by a link capacitance. The nominal skew change can be removed by tuning tapping points as in [31]. An example of the tuning is shown in Figure 12. Even though the location of the tapping point is determined based on Elmore delay model in [31], the basic idea in [31] can be applied on any delay model. As long as a delay model is monotone, the location of the tapping point can be found through a binary search. After tuning tapping points, the link resistance can be inserted. According to the analysis in Section B, link resistance does not affect nominal skew when its two endpoints have zero nominal skew before link insertion.

We add link capacitance to all selected node pairs simultaneously and perform tuning only once at each tapping point before inserting link resistances. The tuning proceed in a bottom-up traversal of the clock tree. During the traversal, when a tapping point is encountered, the tuning is performed in the same way as in [31]. After the tuning is completed, the clock tree should provide skews same as those in the initial tree.

Our algorithm flow can be summarized as:

1. Obtain initial clock tree.
2. Select node pairs where cross links will be inserted.
3. Add link capacitance to the selected nodes.
4. Tune tapping points to restore original skew.
5. Insert link resistance to the selected node pairs.

2. Selecting Node Pairs for Link Insertion

In the algorithm flow, the major problem is how to choose the node pairs for link insertions. We always choose node pairs with zero nominal skew so that no nominal skew is affected by the link resistance. We also prefer node pairs close to each other so that the link capacitance C_l is small and less wire snaking may happen in tuning tapping points. Based on the analysis in Section B, we investigate a rule based selection scheme and a minimum weight matching based selection scheme.

a. Rule Based Selection Scheme

The rule based approach is derived directly from Equation(4.3) and Lemma 2 in Section 2. The conclusions in Section 3 are less rigorous than those in Section 2 and hard to be translated to clear rules. Lemma 2 indicates that the skew variability reduction for a pair of sinks

is more effective when the link is closer to the sinks. Therefore, we restrain the node pairs to be sink pairs for zero skew routing.

α **rule**: In the initial tree, $R_{loop} = R_l + r_u - r_w$ is the total resistance along the loop of $p \rightsquigarrow u \rightsquigarrow w \rightsquigarrow p$. According to Equation(4.3), the link resistance is more effective when the ratio $\alpha = \frac{R_l}{R_{loop}}$ is smaller. Thus the first rule is the ratio $\alpha \leq \alpha_{max}$.

β **rule**: Based on Equation(4.3), the impact of the link capacitance can be reduced when $\beta = |\frac{C_l}{2}(R_{u,u} - R_{w,w})|$ is small or no greater than a certain bound β_{max}. In addition to restraining the link size, this rule is in favor of node pairs that have similar path length from the source.

γ **rule**: The nearest common ancestor(NCA) node for a sink pair has certain depth in the original tree. For the example in Figure 10, the NCA p of r and b has depth 2. We call this depth as the level of the node pair and denote it as γ. Lemma 2 implies that the value of γ needs to be small or no greater than a bound γ_{max}.

It can be observed that there is redundancy in the three rules. But according to our experimental results, all three rules are necessary in general. The rule based node pair selection scheme is simply choosing all the sink pairs that satisfy all the three rules. The advantage of this scheme is its simplicity. The weakness is the neglection of the effects discussed in Section 3 which are considered in the minimum weight based matching algorithm.

b. Minimum Weight Matching Based Selection

According to Lemma 1, when a link resistance is inserted between a pair of nodes, it always reduces skew variation between them. However, the effect of this link on skew of other node pairs is very subtle.

According to the analysis in Section 3, *scenario 3* needs to be avoided, since a link between u and w in a sub-tree T_p may hurt the variability of skew between a sink in T_p and

a sink outside T_p. *Scenario 3* can be avoided by choosing node pairs between left child sub-tree and right child sub-tree of a node of depth 1. For example, links between sub-tree T_p and sub-tree T_d of depth 1 node h in Figure 10 can avoid scenario 3, since there is no sinks outside of T_h. Node pairs for these links can be characterized by the depth of their nearest common ancestor node h, which can be called γ level by using the term in the γ rule of previous section. Therefore, we need to have node pairs with $\gamma = 1$.

Links inserted between sub-tree T_d and sub-tree T_p often improve skew variability between sinks between T_d and T_p according to analysis of *scenario 1*. However, these links may increase skew variability between sinks within T_d or T_p as discussed in *scenario 2*. These degradation can be compensated if links are inserted between sub-sub-trees within sub-tree T_d or T_p. In other words, node pairs of $\gamma = 2$ need to be considered. This procedure can be repeated recursively till γ is sufficiently large. The sub-trees corresponding to large γ are mostly small and the skew variability inside is usually insignificant. The main algorithm description on this recursive procedure is given in Figure 13.

Procedure: $SelectNodePairs(T_v)$
Input: Sub-tree T_v rooted at node v **Output:** Node pair set P
1. $l \leftarrow$ left child node of v
2. $r \leftarrow$ right child node of v
3. $P \leftarrow PairBetweenTrees(T_l, T_r)$
4. If $Depth(v) == DepthLimit$, return P
5. $P \leftarrow P \cup SelectNodePairs(T_l)$
6. $P \leftarrow P \cup SelectNodePairs(T_r)$
7. Return P

Fig. 13. Main algorithm of selecting node pairs for link insertion.

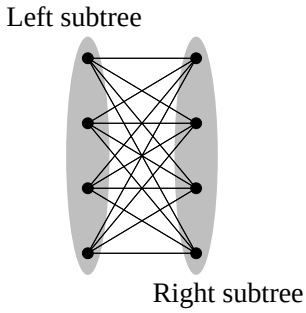

Fig. 14. A bipartite graph model for selecting 4 node pairs between two sub-trees. Each node corresponds to a sub-sub-tree in a sub-tree. An edge weight is the shortest Manhattan distance between leaf(sink) nodes of two sub-sub-trees.

A sub-problem to be solved is how to select node pairs between two sub-trees T_l and T_r. This is the subroutine $PairBetweenTrees(T_l, T_r)$ in line 3 of Figure 13. We decompose each sub-tree into k sub-sub-trees and select k node pairs between sub-sub-trees in T_l and sub-sub-trees in T_r. This problem can be modeled in a bipartite graph and solved by the minimum weight matching algorithm. If the sub-tree T_l is decomposed to sub-sub-tree set $S_l = \{T_{l1}, T_{l2}, ...T_{lk}\}$ and T_r is decomposed to $S_r = \{T_{r1}, T_{r2}, ...T_{rk}\}$, each node in the bipartite graph corresponds to a sub-sub-tree. There is an edge between each node in S_l and each node in S_r. The edge weight between a sub-sub-tree pair T_{li} and T_{rj} is the Manhattan distance between the nearest sink pair $u \in T_{li}$ and $w \in T_{rj}$. An example of the bipartite graph with $k = 4$ is shown in Figure 14. The minimum weight matching result may select a set of node pairs between S_l and S_r with the minimum total link length. The rationale behind this scheme is to distribute the links evenly in a sub-tree such that the *scenario 2* effect is less.

3. Non-zero Skew Routing

The link insertion based non-tree clock routing can be easily extended to achieve non-zero skews. Node pairs can be selected same as in Figure 13. If a sink pair (a,c) have non-zero nominal skew $q_{a,c} = t_a - t_c > 0$, a link can be inserted between sink c and a point k in the parent edge of a such that nominal delay $t_c = t_k$ as illustrated in Figure 10.

D. Experimental Results

The experiments are performed on a Linux machine with dual $1GHz$ AMD microprocessor and 512M memory. The benchmark circuits are r1-r5 downloaded from GSRC Bookshelf $(http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST/)$. The variation factors considered in the experiments include the clock driver resistance, wire width and each sink load capacitance. We let the driver resistance, wire width and the sink capacitance have $\pm 15\%$ variation following a normal distribution. In the experiments, the skew variations and wire-length are compared among clock trees, tree+links and meshes. For each clock network, a Monte Carlo simulation of 1000 trials is performed to obtain the maximum skew variation (MSV) and the standard deviation (SD) of skew variations. A skew variation is the max sink delay minus the min sink delay in a trial.

The clock trees are obtained by running the BST [35] code which is also downloaded from the same web site of GSRC Bookshelf. When running the BST code, the global skew bound is set to 0 so that zero skew clock trees are obtained. The size of benchmark circuits, skew variations and wire-length of clock trees are given in Table II.

We implemented the proposed link insertion based methods and leaf level mesh methods to construct non-tree networks including four variants:

- Link-R: The proposed link insertion based non-tree, with *rule based* node pair selection.

Table II. Maximum skew variation (MSV), standard deviation (SD) and total wire-length of trees. The CPU time is from running BST code.

Testcase	# sinks	MSV	SD	wirelen	CPU(s)
r1	267	0.265	0.042	1320665	1
r2	598	0.759	0.112	2602908	3
r3	862	0.934	0.166	3388951	4
r4	1903	2.321	0.317	6828510	12
r5	3101	5.792	1.149	10242660	18

- Link-M: The proposed link insertion based non-tree, with the *minimum weight matching based* node pair selection.
- Mesh-S: *sparse* leaf level mesh driven by an H-tree.
- Mesh-D: *dense* leaf level mesh driven by an H-tree.

The experimental results on these four variants are shown in Table III. The value of the maximum skew variation(MSV), standard deviation(SD) and wire-length are expressed as the ratio with respect to the results of clock trees.

In *Link-R*, the rules are $\alpha_{max} = 0.1$, $\beta_{max} = 10$ for r1-r3, $\beta_{max} = 50$ for r4 and r5, and $\gamma_{max} = 1$. These rules are chosen empirically as they yield relatively low variation results. In *Link-M*, $k = 2$ for $\gamma = 1$ for r1 and r2, i.e., 2 links are inserted at the level of $\gamma = 1$. Since r3 is larger, $k = 4$ at $\gamma = 1$ is applied. Testcase of r4 and r5 are the largest, hence, we insert links with $k = 2$ at level $\gamma = 2$ in addition to 4 links at $\gamma = 1$.

The observations from Table III include:

- The minimum weight matching based link method is always superior to the rule based method on both variability and wire-length. For this reason, we skip results of rule based method with other rule parameters.
- A dense mesh always yields less variations than a sparse mesh, but consumes more wire-length as expected.
- All four variants work better on larger nets than on smaller nets. For a large net, a tree is dense in term of the number of wire segments, therefore redundant signal

Table III. Skew variations and wire-length in terms of tree results. Size of a tree+link network is the number of links. Size of a mesh is *#rows* $\times$ *#columns*.

Case	Method	size	MSV	SD	wirelen	CPU(s)
r1	Link-R	6	0.65	0.97	1.053	0.039
	Link-M	2	0.60	0.80	1.009	0.068
	Mesh-S	11×11	0.99	0.99	1.69	0.045
	Mesh-D	21×21	0.76	0.66	2.42	0.045
r2	Link-R	20	0.72	0.90	1.027	0.087
	Link-M	2	0.68	0.84	1.009	0.098
	Mesh-S	15×15	0.84	0.82	1.76	0.046
	Mesh-D	29×29	0.60	0.48	2.38	0.046
r3	Link-R	29	0.76	0.88	1.074	0.13
	Link-M	4	0.64	0.83	1.017	0.18
	Mesh-S	19×21	0.26	0.36	1.69	0.046
	Mesh-D	35×37	0.15	0.19	2.47	0.046
r4	Link-R	19	0.53	0.35	1.013	0.38
	Link-M	6	0.46	0.35	1.008	0.43
	Mesh-S	27×29	0.23	0.34	1.68	0.048
	Mesh-D	55×57	0.09	0.18	2.32	0.048
r5	Link-R	57	0.31	0.15	1.030	0.53
	Link-M	6	0.26	0.14	1.008	0.52
	Mesh-S	37×39	0.08	0.10	1.61	0.051
	Mesh-D	75×77	0.03	0.06	2.31	0.051

propagation paths can be established with less efforts. In other words, a link of same size has more chance to short two tree segments in a denser tree (or larger net) than in a sparse tree (or smaller net).

- Except r1, a mesh usually provides lower skew variability than a link based non-tree. But, the wire increase of a mesh is much greater than a link based non-tree. For all solutions from *Link-M*, the wire-length increase over a tree is never greater than 2%.

- The method of *Link-M* results in $32\% - 74\%$ reduction on the maximum skew variation, and $10\% - 86\%$ reduction on the standard deviation, except for r1. Considering less than 2% wire increase, such significant improvement indicates great wire usage efficiency.

The CPU time in seconds are displayed in the right-most column in Table III. The

CPU time for link insertion includes the time for node pair selection and tuning tapping points. Even though the CPU time of link insertion is usually greater than constructing a mesh, it is still negligible, especially compared with the time of clock tree construction.

An experiment on non-zero skew routing is performed on r1. The result shows that *Link-M* reduces standard deviation by 11% over tree with 2% increase on wire-length. A dense mesh reduces standard deviation by 15% with 179% increase on wire-length.

The conclusion and future research for this chapter will be discussed in the chapter V of this thesis.

CHAPTER V

CONCLUSION

In the first part of the thesis, an analytical bound for the unwanted skew due to wire width variation has been obtained. Experimental results show that our method is safer, faster and more accurate than the interval analysis. Since this bound can be obtained very quickly, it can be applied to interconnect variation driven design and design planning.

In the second part of the thesis, a low cost non-tree clock routing method has been proposed to reduce skew variability. The non-tree network is obtained by inserting cross links in a given clock tree. The effect of link insertion on skew variation has been analyzed. Based on the analysis, link insertion algorithms have been developed. Experimental results show that this method can reduce skew variations remarkably with little extra wire resource. This method can be applied to achieve low variation non-zero skew as well. Its accuracy can be improved by adopting a higher order delay model. The efficiency of link insertion can be further improved by considering skew permissible ranges[30] so that links are inserted only between nodes with tight permissible ranges.

This thesis work is one step further toward addressing the challenging skew variation problem.

REFERENCES

[1] S. R. Nassif, "Modeling and analysis of manufacturing variations," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, San Diego, CA, May 2001, pp. 223–228.

[2] 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*, vol.19, no.6, pp.635–644, June 2000.

[3] J. Chung and C.K. Cheng, "Optimal Buffered Clock Tree Synthesis," *IEEE ASIC conference*, Austin, TX, Sept. 1994, pp. 130–133.

[4] Y. Liu, S. R. Nassif, L. T. Pileggi, and A. J. Strojwas, "Impact of interconnect variations on the clock skew of a gigahertz microprocessor," in *Proceedings of the ACM/IEEE Design Automation Conference*, Los Angeles, CA, June 2000, pp. 168–171.

[5] S. Pullela, N. Menezes, and L. T. Pillage, "Reliable non-zero skew clock trees using wire width optimization," in *Proceedings of the ACM/IEEE Design Automation Conference*, Dallas, TX, June 1993, pp. 165–170.

[6] M. P. Desai, R. Cvijetic, and J. Jensen, "Sizing of clock distribution networks for high performance CPU chips," in *Proceedings of the ACM/IEEE Design Automation Conference*, Las Vegas, NV, June 1996, pp. 389–394.

[7] B. Lu, J. Hu, G. Ellis, H. Su, "Process variation aware clock tree routing," *Proceedings of the International Symposium on Physical Design*, Monterey, CA, April 2003, pp. 174–181.

[8] S. Lin and C. K. Wong, "Process-variation-tolerant clock skew minimization," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, November 1994, pp. 284–288.

[9] H. Su and S. S. Sapatnekar, "Hybrid structured clock network construction," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, November 2001, pp. 333–336.

[10] T. Xue and E. S. Kuh, "Post routing performance optimization via multi-link insertion and non-uniform wiresizing," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*,San Jose, CA, November 1995, pp. 575–580.

[11] P. K. Chan and K. Karplus, "Computing signal delay in general RC networks by tree/link partitioning," *IEEE Transactions on Computer-Aided Design*, vol.9, no.8, pp. 898–902, August 1990.

[12] C. L. Harkness and D. P. Lopresti, "Interval methods for modeling uncertainty in RC timing analysis," *IEEE Transactions on Computer-Aided Design*, vol.11, no.11, pp. 1388–1401, November 1992.

[13] A. D. Fabbro, B. Franzini, L. Croce, and C. Guardiani, "An assigned probability technique to derive realistic worst-case timing models of digital standard cells," in *Proceedings of the ACM/IEEE Design Automation Conference*, San Francisco, CA, June 1995, pp. 702–706.

[14] N. Chang, V. Kanevsky, O. S. Nakagawa, K. Rahmat, and S.-Y. Oh, "Fast generation of statistically-based worst-case modeling of on-chip interconnect," in *Proceedings of the IEEE International Conference on Computer Design*, Austin, TX, October 1997, pp. 720–725.

[15] P. Zarkesh-Ha, T. Mule, and J. D. Meindl, "Characterization and modeling of clock skew with process variations," in *Proceedings of the IEEE Custom Integrated Circuits Conference*, San Diego, CA, May 1999, pp. 441–444.

[16] D. Sylvester, O. S. Nakagawa, and C. Hu, "Modeling the impact of back-end process variation on circuit performance," in *Proceedings of the International Symposium on VLSI Technology, Systems and Applications*, Taipei, Taiwan, June 1999, pp. 58–61.

[17] E. Acar, S. R. Nassif, Y. Liu, and L. T. Pileggi, "Assessment of true worst case circuit performance under interconnect parameter variations," in *Workshop Notes, International Workshop on Timing Issues in the Specification and Synthesis of Digital Systems*, Austin, TX, December 2000, pp. 45–49.

[18] S. Zanella, A. Nardi, A. Neviani, M. Quarantelli, S. Saxena, and C. Guardiani, "Analysis of the impact of process variations on clock skew," *IEEE Transactions on Semiconductor Manufacturing*, vol.13, no.4, pp. 401–407, November 2000.

[19] M. Orshansky and K. Keutzer, "A general probabilistic framework for worst case timing analysis," in *Proceedings of the ACM/IEEE Design Automation Conference*, New Orleans, LA, June 2002, pp. 556–561.

[20] P. J. Restle, T. G. McNamara, D. A. Webber, P. J. Camporese, K. F. Eng, K. A. Jenkins, D. H. Allen, M. J. Rohn, M. P. Quaranta, D. W. Boerstler, C. J. Alpert, C. A. Carter, R. N. Bailey, J. G. Petrovick, B. L. Krauter, and B. D. McCredie, "A clock distribution network for microprocessors," *IEEE Journal of Solid-State Circuits*, vol.36, no.5, pp. 792–799, May 2001.

[21] N. A. Kurd, J. S. Barkatullah, R. O. Dizon, T. D. Fletcher, and P. D. Madland, "A multigigahertz clocking scheme for the Pentium 4 microprocessor," *IEEE Journal of Solid-State Circuits*, vol.36, no.11, pp. 1647–1653, November 2001.

[22] J. L. Neves and E. G. Friedman, "Topological design of clock distribution networks based on non-zero clock skew specifications," in *Proceedings of the Midwest Symposium on Circuits and Systems*, Detroit, MI, August 1993, pp. 468–471.

[23] J. G. Xi and W. W.-M. Dai, "Useful-skew clock routing with gate sizing for low power design," *Journal of VLSI Signal Processing*, vol.16, no.(2/3), pp. 163–179, Jun./Jul. 1997.

[24] C.-W. A. Tsao and C.-K. Koh, "UST/DME: a clock tree router for general skew constraints," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, November 2000, pp. 400–405.

[25] J. P. Fishburn, "Clock skew optimization," *IEEE Transactions on Computers*, vol.39, no.7, pp. 945–951, July 1990.

[26] W.-C. D. Lam, C.-K. Koh, and C.-W. A. Tsao, "Power supply noise suppression via clock skew scheduling," in *Proceedings of the IEEE International Symposium on Quality Electronic Design*, San Jose, CA, March 2002, pp. 355–360.

[27] J. P. Fishburn and C. A. Schevon, "Shaping a distributed RC line to minimize Elmore delay," *IEEE Transactions on Circuits and Systems*, vol.42, no.12, pp. 1020–1022, December 1995.

[28] C.-P. Chen and D. F. Wong, "Optimal wire-sizing function with fringing capacitance consideration," *Technical Report TR96-28*, Department of Computer Science, The University of Texas, Austin, November, 1996.

[29] C.-P. Chen, H. Zhou, and D. F. Wong, "Optimal non-uniform wire-sizing under the Elmore delay model," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, November 1996, pp. 38–43.

[30] J. L. Neves and E. G. Friedman, "Optimal clock skew scheduling tolerant to process variations," in *Proceedings of the ACM/IEEE Design Automation Conference*, Las Vegas, NV, June 1996, pp. 623–628.

[31] R.-S. Tsay, "Exact zero skew," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, Santa Clara, CA, November 1991, pp. 336–339.

[32] T.-H. Chao, Y.-C. Hsu, J.-M. Ho, K. D. Boese, and A. B. Kahng. "Zero skew clock routing with minimum wirelength," *IEEE Transactions on Circuits and Systems - Analog and Digital Signal Processing*, vol.39, no.11, pp.799–814, November 1992.

[33] K. D. Boese, A. B. Kahng, B. A. McCoy, and G. Robins, "Near-optimal critical sink routing tree constructions," *IEEE Transactions on Computer-Aided Design*, vol.14, no.12, pp. 1417–1436, December 1995.

[34] J. Qian, S. Pullela, and L. T. Pillage, "Modeling the effective capacitance for the RC interconnect of CMOS gates," *IEEE Transactions on Computer-Aided Design*, vol.13, no.12, pp. 1526–1535, December 1994.

[35] J. Cong, A. B. Kahng, C.-K. Koh and C.-W. A. Tsao, "Bounded-skew clock and Steiner routing under Elmore delay," in *Proceedings of the IEEE/ACM International Conference on Computer-Aided Design*, San Jose, CA, November 1995, pp. 66–71.

[36] J. Cong and K.-S. Leung, "Optimal wiresizing under the distributed Elmore delay," *IEEE Transactions on Computer-Aided Design*, vol.14, no.3, pp.321–336, June 1995.

VITA

Anand Kumar Rajaram was born on January 6^{th} 1981 in Pammal, a nice suburban city of Chennai(Madras), India. He completed his entire schooling in Sri Shankara Vidhyalaya, Pammal, and later he joined the College of Engineering Guindy, Anna University in August 1998. He earned his Bachelor of Engineering degree from Anna University with distinction in May 2002. He moved to Texas A&M University in the Fall of 2002 where he earned his Master of Science degree in Electrical Engineering.

The typist for this thesis was Anand Rajaram.