Buffered Clock Tree Synthesis for 3D ICs Under Thermal Variations

Jacob Minz[†], Xin Zhao, and Sung Kyu Lim
[†]Synopsys Corporation, Mountain View, California, USA
Georgia Institute of Technology, Atlanta, Georgia, USA
jacob.minz@synopsys.com, xinzhao@ece.gatech.edu, limsk@ece.gatech.edu

***Abstract*— In this paper, we study the buffered clock tree synthesis problem under thermal variations for 3D IC technology. Our major contribution is the Balanced Skew Theorem, which provides a theoretical background to efficiently construct a buffered 3D clock tree that minimizes and balances the skew values under two distinct non-uniform thermal profiles. Our clock tree synthesis algorithm named BURITO (Buffered Clock Tree With Thermal Optimization) first constructs a 3D abstract tree under the wirelength vs via-congestion tradeoff. This abstract tree is then embedded, buffered, and refined under the given non-uniform thermal profiles so that the temperature-dependent skews are minimized and balanced. Experimental results show that our algorithms significantly reduce and perfectly balance clock skew values with minimal wirelength overhead.**

I. Introduction

The 3D ICs [1] have recently demonstrated a huge potential in reducing wirelength and area of a design, thereby improving performance and cost. However, its widespread adoption has been hampered primarily by thermal issues. The substrates experience higher thermal gradients due to multiple active layers plus the use of insulating dielectric layers of low thermal conductivity. These higher thermal variations induce a substantial amount of skew in the clock tree, which has adverse implications for the performance and reliability of 3D ICs. The goal of this paper is to present the first algorithm for the buffered clock tree synthesis under thermal variations for 3D IC technology. We are specifically targeting the face-to-face wafer-bonded 3D stacked IC technology [1]. Through-vias have a relatively small cost in the face-to-face bonded 3D ICs. Thus, 3D clock routing may include many through vias as shown in Figure 1.

The history of clock tree synthesis under thermal variation is short. TACO [2] suggests to construct a tree that balances the skew under the two given static thermal profiles (= uniform and worst). The argument is that the skew optimized under one thermal profile will vary significantly under the other profile, thus the need to balance. TACO first builds a non-buffered, zero-skew clock tree under a uniform thermal profile and refines it under worst-case (= non-uniform) profile. Since the thermal gradient is changing over time while the chip is running, a transient analysis/optimization is an extremely difficult task, if not impossible. Thus, we believe that bounding the skew between the two extreme cases is much more practical. The authors in [3] extended [2] by considering bounded-skew. However, these works do not guarantee a perfect skew balance between the two thermal profiles given. This means that the skew values under two different profiles may differ by a non-trivial margin. In addition, it is not clear how these heuristics can be extended to consider buffer insertion or merging of two clock nodes in two different dies as in 3D ICs.

*This material is based upon work supported by the National Science Foundation under CAREER Grant No. CCF-0546382 and Focus Center for Circuit & System Solutions (C2S2), Semiconductor Research Corporation.

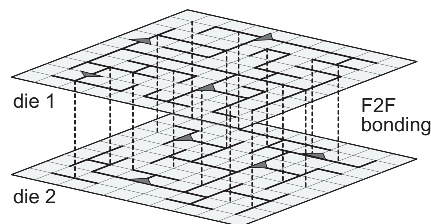

Fig. 1. 3D buffered clock tree with face-to-face bonding. Dotted lines denote through-vias.

The contribution of our work is as follows:

- We present the Balanced Skew Theorem, which provides a theoretical background on the efficient construction of a buffered 3D clock tree that perfectly balances the skew values under two distinct non-uniform thermal profiles. This theorem simplifies the clock tree construction under thermal consideration considerably and makes it easier to handle buffers and through-vias.
- We develop the first buffered clock tree synthesis algorithm under thermal variations for 3D IC technology. Our 3D clock trees are built under the through-via bound set for the face-to-face bonding technology and achieve a smooth tradeoff between through-via and wirelength.

Experimental results show that our algorithms obtain comparable results to the TACO [2] for 2D clock trees. In case of 3D clock trees, our algorithm significantly reduces and perfectly balances clock skews for both non-buffered and buffered clock trees with minimal wirelength overhead.

II. Problem Formulation

The following are the inputs to the *Thermal-aware 3D Buffered Clock Tree Synthesis Problem*: (i) two dies bonded in face-to-face, (ii) two sets of sinks S_1 from die 1 and S_2 from die 2, and their locations and input capacitances, (iii) through-via resistance r_v, capacitance c_v, and bound B, (iv)

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

two steady state thermal profiles T_u (covering both dies) and T_w (covering both dies), and (v) buffer input capacitance c_b, output resistance r_b and maximum load C_{max}. The goal is to construct a 3D buffered tree T that connects the sinks in $S_1 \cup S_2$ while satisfying the following constraints: (i) the number of through-vias used does not exceed B, (ii) the distribution of through-via across the dies is even, (iii) the distribution of buffers inserted in both dies is even, and (iv) each buffer drives a load up to C_{max}. The objective is to minimize and balance the skew between T_u and T_w and minimize the total wirelength.

Thermal distribution keeps changing during the circuit operation, so it is impossible to tackle all possible distributions. Thus, we target two representative profiles, namely, uniform and worst-case, so that the skew values are identical under these two situations. Note that a zero skew tree under uniform thermal profile will experience a significant skew change under the worst-case profile. Likewise, a zero skew tree under the worst-case profile will incur a significant skew under the uniform profile. This is why we are interested in balancing (and minimizing) the skew under these two profiles so that the skew values are likely to be bounded within a range.

We restrict our buffered 3D clock trees in the following ways.

- The two dies that are face-to-face bonded have the same wire parasitics (= unit length resistance and capacitance).
- The merging point u and the through-via are co-located as shown in Figure 2(a).
- For any given internal node with a buffer attached, we do not separate the buffer-node pair during the thermal optimization (= internal node relocation).

Our wire delay computation as based on Elmore model.[1] We use the recent temperature dependent wire delay model proposed in [4], [5]. The line resistance per unit length can be calculated as

$$r(x) = r_0(1 + \beta \cdot T(x))$$

where r_0 is the resistance at $0°C$, β is the temperature coefficient of resistance, and $T(x)$ denotes the temperature at location x. In our model the through-via resistance and buffer output resistance also vary linearly with temperature. We assume, however, that the wire capacitance and buffer input capacitance do not as suggested in [4], [5]. We also assume that the intrinsic buffer delay does not vary with temperature. In addition, we do not model the Joule heating of the wires.

III. Thermal-aware Skew Analysis

A. Skew Gradient

Table I shows the notations and definitions used in the following sections. In order to compute the interconnect delay under non-uniform thermal profile, we first segment the interconnect, compute the temperature dependent delay for each segment, and sum them up. To facilitate this approach, we define the *skew gradient* as the rate of change of skew function along the path connecting two clock nodes. The Elmore delay at the merging point u, denoted $d(u,p)$, is:

TABLE I
Variables and constants used in our formulation

Symbol	Description
p, q	root nodes of two sub-trees merging at node u.
L	distance between p and q.
t_p	max Elmore delay between p and its sink nodes.
C_p	downstream capacitance at node p.
$c(x)$	downstream capacitance at point x ($p \leq x \leq q$).
$r_v(x)$, c_v	through-via parasitic resistance (temperature dependent) and capacitance.
$r(x)$, c	unit-length wire resistance (temperature dependent) and capacitance.
$r_d(x)$, t_d, c_d	buffer driving resistance (temperature dependent), intrinsic delay, and input capacitance.
T_u, T_w	two input non-uniform thermal profiles.
$d(u,p)$	Elmore delay from u to the sink nodes in the subtree rooted at p.
$s(u)$	skew at merging point, which is computed by $d(u,p) - d(u,q)$.
$s(u,T)$	skew at merging point u under thermal profile T.

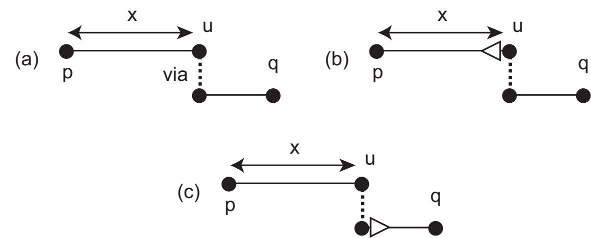

Fig. 2. Side-view of 3D merging of two subtrees rooted at p (= in the top layer) and q (= in the bottom layer) at a merging point u (= in the top layer). (a) non-buffered clock tree merging, (b) buffer inserted on $e(u,p)$, (c) buffer inserted on $e(u,q)$.

$$d(u,p) = t_p + \int_0^x r(z)\big(C_p + \int_z^x c(y)dy\big)dz \quad (1)$$

$d(u,q)$ can be defined similarly. The skew at u is

$$\begin{aligned} s(u) \quad = \quad & t_p - (t_q + \int_0^L r(z)(C_q + c(L-z))dz) \\ & + \int_0^x r(z)(C_p + C_q + cL)dz \end{aligned} \quad (2)$$

Thus, the skew gradient is

$$\frac{ds}{dx} = (C_p + C_q + cL)r(x) \quad (3)$$

Note that the skew gradient is linearly proportional to $r(x)$, the temperature-dependent resistance. Finally, the skew at point u under a non-uniform thermal profile T is

$$s(u,T) = s(0) + \sum_{i=1}^{N} \frac{ds(i)}{dx}\Delta x_i \quad (4)$$

The wire from p to q is segmented into N parts by the same pitch (= Δx_i) as the underlying thermal grid.

[1]Our skew analysis and theorems presented in this paper can be extended easily to a more general model, where the delay is linearly dependent on wire/via/buffer resistance.

B. Non-Buffered 3D Clock Tree Merging

Figure 2(a) illustrates non-buffered clock tree merging. The Elmore delay at u after adding $e(u,p)$ is:

$$d(u,p) = r(x)x(\frac{cx}{2} + C_p) + t_p \quad (5)$$

The Elmore delay at u after adding $e(u,q)$, i.e., $d(u,q)$ is

$$r_v(x)(\frac{c_v}{2} + c(L-x) + C_q) + r(x)(L-x)(\frac{c(L-x)}{2} + C_q) + t_q \quad (6)$$

Obtaining $s(x) = d(u,p) - d(u,q)$ and differentiating it, we get

$$\frac{ds(x)}{dx} = r(x)(C_p + C_q + cL) + r_v(x)c - \big(C_q + c(L-x) + \frac{c_v}{2}\big)\frac{dr_v(x)}{dx} \quad (7)$$

Note that the unit-length resistance $r(x)$ and through-via resistance $r_v(x)$ are functions of x. In addition, $s(0)$ is given by:

$$s(0) = t_p - \big(r_v(x)(\frac{c_v}{2} + cL + C_q) + r(x)L(\frac{cL}{2} + C_q) + t_q\big) \quad (8)$$

We compute the skew function using Equation (4), (7), and (8).

C. Buffered 3D Clock Tree Merging

In case of buffered tree merging, we consider the two cases shown in Figure 2(b) and 2(c). The buffer is located either in the top layer (= Figure 2(b)) or in the bottom layer (= Figure 2(c)). In case of the former, the skew gradient is given by:

$$\frac{ds(x)}{dx} = r(x)(C_p + C_q + cL) + \big(r_d(x) + r_v(x)\big)c + (C_p + cx)\frac{dr_d(x)}{dx} - (\frac{c_v}{2} + c(L-x) + C_q)\frac{dr_v(x)}{dx} \quad (9)$$

Note that $r(x)$, $r_d(x)$ and $r_v(x)$ are all functions of x. In addition, $s(0)$ becomes

$$t_p + t_d + r_d(x)C_p - r_v(x)(\frac{c_v}{2} + cL + C_q) - r(x)L(\frac{cL}{2} + C_q) - t_q \quad (10)$$

Lastly, the skew at point u under the non-uniform thermal profile is computed using Equation (4). The skew gradient and $s(0)$ equations for other case, i.e., the buffer is located in the bottom layer, can be derived similarly.

D. Balanced Skew Theorem

Assume that two clock tree nodes p and q are merged at node u. Under the two given thermal profiles T_u and T_w, node p is associated with the following delay values: (1) $t_p^{min}(T_u)$: minimum delay at p under T_u, (2) $t_p^{max}(T_u)$: maximum delay at p under T_u, (3) $t_p^{min}(T_w)$: minimum delay at p under T_w, (4) $t_p^{max}(T_w)$: maximum delay at p under T_w. Similar delay values also exist for q. Let

$$t_p^a(T_u) = \frac{t_p^{min}(T_u) + t_p^{max}(T_u)}{2} \quad (11)$$

denote the average delay for p under T_u. We define $t_p^a(T_w)$, $t_q^a(T_u)$, and $t_q^a(T_w)$ similarly. Clearly,

$$s(p,T_u) = |t_p^{max}(T_u) - t_p^{min}(T_u)| \quad (12)$$
$$s(q,T_u) = |t_q^{max}(T_u) - t_q^{min}(T_u)| \quad (13)$$

Similar properties hold under T_w. The balanced skew point is defined as follows:

Definition 1: Balanced Skew Point—A clock merging point x is a balanced skew point if $s(x,T_u) = s(x,T_w)$.

Let $S(t_p,t_q,u)$ denote the skew at a point u, where two clock tree nodes p and q are merged (t_p and t_q are the delay at p and q).

Lemma 1: For any constant k, the following expressions hold:

$$S(t_p + k, t_q, u) = S(t_p, t_q, u) + k \quad (14)$$
$$S(t_p, t_q + k, u) = S(t_p, t_q, u) - k \quad (15)$$

Proof: The definition of skew is $S(t_p,t_q,u) = d(u,p) - d(u,q)$. In case t_p is increased by k, $d(u,p)$ is also increased by k. This causes the overall skew to increase by k, which proves the first property. In case t_q is increased by k, $d(u,q)$ is also increased by k. This causes the overall skew to decrease by k, which proves the second property. ■

The following theorem provides an efficient way to compute balanced skew points for a given pair of clock trees:

Theorem 1: If the skew at node p and q are balanced under T_u and T_w, then the balanced skew merging point z can be calculated by solving the following:

$$S(t_p^a(T_u), t_q^a(T_u), z) + S(t_p^a(T_w), t_q^a(T_w), z) = 0 \quad (16)$$

The proof is omitted due to the space limit. The significance of Theorem 1 is that the balanced skew point for any given pair of subtrees can be computed given that the root nodes of the subtrees are skew-balanced. This enables the well-known dynamic programming approaches such as ZST [6] and DME [7] to easily adopt our method to compute balanced skew points. We now present our Balanced Skew Theorem:[2]

Theorem 2: The construction of a clock tree (buffered or non-buffered) with balanced skew under thermal profiles T_1 and T_2 is equivalent to constructing a zero-skew tree under profile $T_a = (T_1 + T_2)/2$, where the temperature of each grid in T_a is the average between T_1 and T_2 of the corresponding location.

The proof is omitted due to the space limit. The significance of Theorem 2 is that it simplifies the clock tree construction under thermal variations considerably because the clock tree now has to be optimized for one thermal profile instead of two. Without Theorem 2, one has to calculate two different skew functions under two thermal profiles, and the balanced skew point is the intersection the two curves.[3]

[2] This theorem can easily be applied to 2D ICs as well by setting $r_v = c_v = 0$.

[3] Skew Balance under more than two thermal profiles is not well defined since the skew curves may not merge at a single point depending on the thermal distributions given.

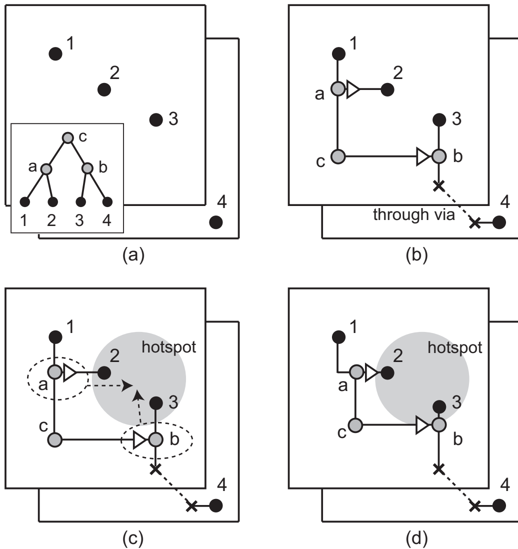

Fig. 3. Overview of our 4-step approach. (a) abstract tree generation, (b) embedding and buffering, (c) thermal analysis, (d) thermal-aware optimization.

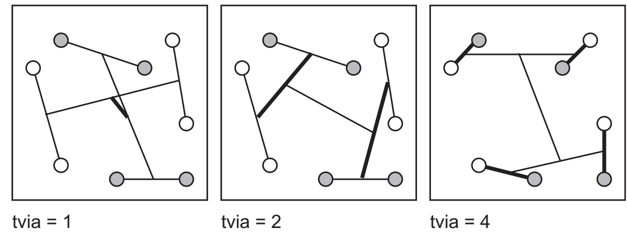

Fig. 4. 3D abstract trees constructed with our 3D-MMM algorithm under various through-via constraints. White and gray nodes are respectively in the top and bottom layer. Thick lines denote connections that need through-vias. The overall wirelength decreases as more vias are used in the trees.

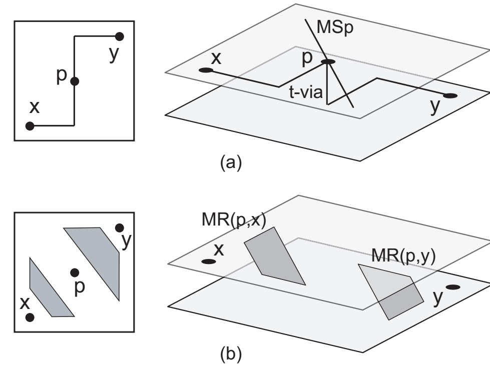

Fig. 5. (a) 3D merging segment for unbuffered tree, (b) 3D merging region for buffered tree

IV. 3D Clock Routing Algorithm

Due to the complexity of buffered 3D clock tree synthesis problem under thermal variations, we adopt a four-step approach: 3D abstract tree generation, embedding and buffering, thermal analysis [8], and thermal-aware optimization. An illustration is shown in Figure 3.

A. 3D Abstract Tree Generation

The goal of 3D abstract tree generation is to facilitate an even distribution of through-vias while minimizing the wirelength under through-via bound. The MMM algorithm [9] has been extended to generate the abstract merging tree for the 3D clock sinks in a top-down manner. Figure 4 shows an illustration. The basic idea is to recursively divide the given sink set into two subsets until each sink belongs to its own set. We then visit each sink in a bottom-up fashion and start merging subtrees until all sinks are connected via a single tree. At each recursive partitioning step, we divide the given set of sinks into two subsets A and B. We consider the following two cases based on the through-via bound set for the current sink set:

- If the via bound is one, the given sink set is partitioned such that the sinks from the same layer belong to the same subset. The connection between A and B needs one through-via.
- If the bound is larger than one, the set is flattened to 2D (z-dimension is ignored) and partitioned geometrically by a horizontal or vertical line. Since each subset contains sinks from both dies, we potentially need many through-vias to connect them.

At the end of partitioning, we decide the through-via bound for each subset as follows: (i) estimate the number of through-vias required by each set and (ii) divide the given bound according to the ratio of estimated vias. The cut direction is determined such that the via bound is balanced in both subsets. The complexity of the algorithm is $O(nlogn)$, where n is the number of sink nodes.

B. Clock Tree Embedding and Buffering

During the embedding and buffering step, the internal nodes of the 3D abstract tree are placed and buffers are inserted under zero-skew constraint and uniform thermal distribution. The classic DME algorithm [7] is extended to generate topology embedding for the given 3D abstract tree. We extend [10] to perform buffer insertion in 3D environment. Figure 5 illustrates how (i) the merging segments for unbuffered merging, and (ii) the merging region for buffered merging are generated in 3D cases. Our work inserts buffer in the given tree by using a cost function that considers even distribution (= intra and inter-layer distribution) of buffers and wirelength. The main difference between our approach and [10] is that our approach works on a merging tree while the latter directly generates merging segments by considering buffers in the merging cost, thereby bypassing abstract tree generation. While the latter approach guarantees better quality solutions, it suffers from prohibitive complexity (= $O(n^3)$). The complexity of our approach is $0(n)$.

C. Thermal-aware Clock Tree Optimization

The next step is to perform thermal analysis on the buffered clock tree we obtain from the previous steps and optimize it based on the temperature calculation. The goal of our

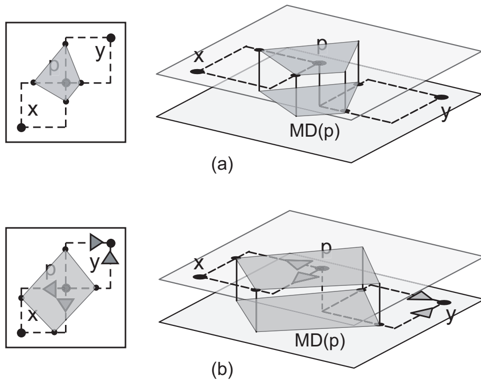

Fig. 6. (a) 3D merging diamond for unbuffered tree under thermal variation, (b) 3D merging diamond for buffered tree.

TABLE III
RESULTS FOR 3D ABSTRACT TREE GENERATION

	Via Bound=1		Via Bound=10%			Via Bound=100%		
	wire	dly	wire	dly	vias	wire	dly	vias
r1	1934487	1.3	1680766	2.4	27	1497192	2.2	97
r2	4148464	3.7	3535557	6.2	60	2996564	5.9	217
r3	5319061	6.1	4469137	9.5	87	3870671	9.1	306
r4	10790865	15.6	9146692	28.9	191	7785551	26.9	716
r5	15849880	28.9	13235081	50.0	310	11388363	48.6	1135
R	1.00	1.00	0.85	1.72	1.00	0.73	1.63	3.63

thermal-aware optimization step is to relocate the internal clock nodes and buffers for skew minimization and balancing under thermal variations.[4] We extend the merging diamond concept [2] to handle the merging of two clock nodes located in different dies. An illustration is shown in Figure 6. In this case, the balanced skew points for each of the 4 L-shaped paths is determined by the Balanced Skew Theorem developed in Section III-D instead of the heuristics used in [2]. We use the term *options* to describe the set of points stored at each internal node, along with the topologies and its child options. The algorithm generates options at each internal node by a bottom-up traversal. The number of options are bounded at each node by solution pruning, where the weighted sum of skews and wirelength is used as the cost function. The options with the best cost are retained while the others are discarded since the other options are less likely to improve either skew or wirelength of the final tree. The optimized tree is obtained by recursively selecting the left and the right options in a top-down traversal. The selection determines the parent-to-child topological path as well as the new location of the internal nodes and buffers. The complexity of the algorithm is $O(n)$, where n is the number of sink nodes.

V. EXPERIMENTAL RESULTS

Our algorithm BURITO was implemented using C++/STL and run on Linux Xbeowulf cluster. The IBM benchmark suite $r1$-$r5$ from [11] was used for the experiments. The 3D sink placement was generated by randomly partitioning the clock sinks into two layers and using an area compression factor of 1. The technology parameters used were $r = 0.003\Omega/\mu m$, $c = 2.0$ x $10^{-17}F/\mu m$, and $\beta = 0.0068/^oC$. The buffer parameters were $r_d = 122\Omega$, $c_d = 400fF$, $t_d = 75ps$, and $C_{max} = 4pF$. The 3D via parameters are $r_v = 0.053\Omega$ and $c_v = 27.9fF$. We use two thermal profiles T-uni (= uniform distribution) and T-worst (non-uniform, worst-case distribution) as done in TACO [2] although our Balanced Skew Theorem handles any two thermal profiles.

Table II first compares the results of TACO [2] and BURITO-2D (= 2D version of BURITO). Both algorithms use the same initial zero-skew clock tree built under a uniform thermal distribution by BST-DME [11]. In this case the only difference is on how the thermal optimization is done on this initial tree. Since the thermal profiles used by TACO are randomly generated and not available, we generated ours based on the same min/max/average temperatures reported by TACO. Therefore, a fair comparison between TACO and BURITO-2D is not possible. In any case, we achieved a slightly better skew improvement. The skews in the optimized trees under uniform and non-uniform temperatures have been perfectly balanced in our approach. TACO does not guarantee balanced skews. Next, the delay improved by 37% at the cost of 13% increase in wirelength after buffer insertion. In addition, skew values also reduced by 13%. Buffers improve the slew rate at the sinks [10], which is measured by the insertion delays, i.e., lower insertion delay means better slew rate.

Table III reports our 3D abstract tree generation results. The baseline was the clock-routing with one through-via. The wirelength was calculated after running DME [7] on the abstract tree. We first note the wirelength vs via usage tradeoff. As the via bound increases, wirelength decreases while via usage increases. Next, the delay increases from bound 1 to bound 10% is primarily due to the via delay itself. However, the delay reduces slightly from 10% to 100% case, which is primarily due to the overall wirelength saving. Figure 7 first shows the wirelength vs via tradeoff. In addition, it shows that the through-vias are evenly placed across the chip.

The results of 3D buffered clock tree optimization with our BURITO-3D are presented in Table IV. The initial clock trees are generated by setting the via bound to 10% of the clock sinks. First, the skew values are larger and unbalanced between T-uni and T-worst in the initial trees, but they are perfectly balanced and smaller in the optimized trees. Second, skew reduction of about 56% was achieved after the optimization at the cost of negligible wirelength increase. Lastly, our thermal optimization had little impact on the delay results.

VI. CONCLUSIONS

This paper studied the buffered clock tree synthesis problem under thermal variations for 3D IC technology. We presented the Balanced Skew Theorem, which enables an efficient construction of a buffered 3D clock tree that minimizes and

[4]Our related experiment suggested that the impact of this internal node/buffer relocation on the underlying thermal profile is minimal due to the small change introduced into the layout. Thus, we assume that the thermal profile remains fixed. Our thermal optimization, however, can be iterated if desired.

TABLE II

COMPARISON BETWEEN TACO AND BURITO-2D. BOTH ALGORITHMS USE THE INITIAL CLOCK TREES GENERATED BY BST-DME. THE DELAY VALUES ARE IN ns, SKEWS IN ps, AND WIRELENGTH IN μm.

	TACO [2]					BURITO-2D (non-buffered)					BURITO-2D (buffered)				
	T-uni		T-worst			T-uni		T-worst			T-uni		T-worst		
ckt	dly	**skew**	dly	**skew**	wire	dly	**skew**	dly	**skew**	wire	dly	**skew**	dly	**skew**	wire
r1	1.8	**23.1**	1.9	**11.4**	1323174	1.8	**14.8**	2.0	**14.8**	1322793	2.2	**8.9**	2.7	**8.9**	1476646
r2	4.5	**86.2**	4.9	**75.7**	2615431	4.4	**99.4**	5.2	**99.4**	2609009	3.7	**121.6**	4.2	**121.6**	2905106
r3	7.1	**107.9**	8.1	**107.9**	3399060	7.0	**78.9**	8.3	**78.9**	3395771	4.3	**106.6**	5.1	**106.6**	3849891
r4	14.7	**220.7**	17.6	**220.7**	6845621	14.7	**367.5**	17.7	**367.5**	6844030	4.9	**246.6**	6.0	**246.6**	7875873
r5	35.9	**629.1**	44.7	**675.1**	10302650	35.7	**379.3**	44.8	**379.3**	10265969	6.4	**303.4**	7.9	**303.4**	11724289
RATIO	1.00	1.00	1.00	1.00	1.00	0.99	0.96	1.03	1.11	0.99	0.63	0.87	0.68	0.98	1.13

TABLE IV

BURITO-3D CLOCK ROUTING RESULTS. THE DELAY VALUES ARE IN ns, SKEWS IN ps, AND WIRELENGTH IN μm.

			initial tree					optimized tree				
			T-uni		T-worst			T-uni		T-worst		
ckt	#sink	#buf	dly	skew	dly	skew	wire	dly	skew	dly	skew	wire
r1(267)	267	15	2.80	0.00	3.25	220.55	1960406	2.83	89.01	3.18	89.01	1957771
r2(598)	598	31	4.08	0.00	4.97	359.03	4112598	4.24	146.70	4.94	146.70	4117089
r3(862)	862	47	4.24	0.00	5.14	500.13	5409447	4.31	243.31	4.96	243.31	5411678
r4(1903)	1903	107	5.36	0.00	6.81	617.95	11500018	5.69	294.26	6.93	294.26	11506562
r5(3101)	3101	127	7.30	0.00	9.56	965.21	16155680	7.50	406.34	9.31	406.34	16172733
RATIO			1.00	-	1.00	1.00	1.00	1.03	-	0.98	0.44	1.00

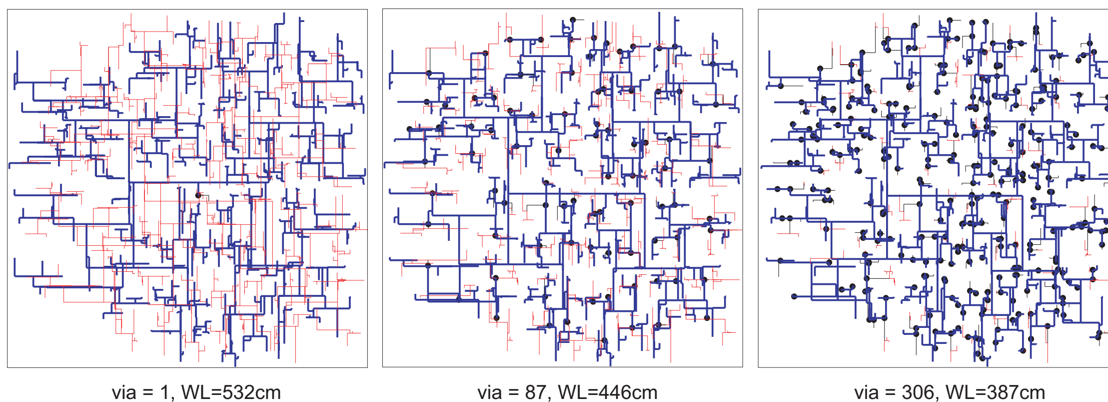

Fig. 7. 3D clock routing for $r3$ with through-via bound of (a) one, (b) 10% of sinks, (c) 100% of sinks. The dots denote the through-vias. The thick and thin lines are routing in two layers.

perfectly balances the skew values under two distinct non-uniform thermal profiles. We developed the first buffered clock tree synthesis algorithm under thermal variations for 3D IC technology. Experimental results show that our algorithms significantly reduce and perfectly balance clock skews with minimal wirelength overhead.

REFERENCES

[1] R. Reif and et al, “Fabrication technologies for three-dimensional integrated circuits,” in *Proc. Int. Symp. on Quality Electronic Design*, 2002.

[2] M. Cho, S. Ahmed, and D. Z. Pan, “TACO: Temperature aware clock optimization,” in *Proc. IEEE Int. Conf. on Computer-Aided Design*, November 2005.

[3] A. Chakraborty, P. Sithambaram, K. Duraisami, A. Macii, E. Macii, and M. Poncino, “Thermal resilient bounded-skew clock tree optimization methodology,” in *Proc. Design, Automation and Test in Europe*, 2006.

[4] A. Ajami, K. Banerjee, and M. Pedram, “Effects of non-uniform substrate temperature on the clock signal integrity in high performance designs,” in *Proc. of IEEE Custom Integrated Circuits Conference*, May 2001, pp. pp. 233–236.

[5] K. Banerjee, S. J. Souri, P. Kapur, and K. Saraswat, “3D ICs: A novel chip design for improving deep submiscron interconnect performance and systems-on-chip integration,” in *Proc. IEEE, Special Issue on Interconnects,*, May 2001, pp. 602–633.

[6] R. S. Tsay, “An exact zero-skew clock routing algorithm,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 1993.

[7] K. D. Boese and A. B. Kahng, “Zero-skew clock routing trees with minimum wirelength,” in *Proc. IEEE Intl. Conf. on ASIC*, 1992, pp. pp. 1.1.1–1.1.5.

[8] T.-Y. Wang and C. C.-P. Chen, “3-d thermal-adi: A linear-time chip level transient thermal simulator,” *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, pp. 1434–1445, 2002.

[9] M. Jackson, A. Srinivasan, and E. Kuh, “Clock routing. for high-performance ICs,” in *Proc. ACM Design Automation Conf.*, 1990.

[10] A. Vittal and M. Marek-Sadowska, “Power optimal buffered clock tree design,” in *Proc. ACM Design Automation Conf.*, 1995, pp. 497–502.

[11] `http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST`.