Low-Power Clock Tree Design for Pre-Bond Testing of 3-D Stacked ICs

Xin Zhao, *Student Member, IEEE,* Dean L. Lewis, *Student Member, IEEE,* Hsien-Hsin S. Lee, *Senior Member, IEEE,* and Sung Kyu Lim, *Senior Member, IEEE*

***Abstract*—Pre-bond testing of 3-D stacked integrated circuits (ICs) involves testing each individual die before bonding. The overall yield of 3-D ICs improves with pre-bond testability because manufacturers can avoid stacking defective dies with good ones. However, pre-bond testability presents unique challenges to 3-D clock tree design. First, each die needs a complete 2-D clock tree to enable pre-bond test. Second, the entire 3-D stack needs a complete 3-D clock tree for post-bond test and operation. In the case of a two-die stack, a straightforward solution is to have two complete 2-D clock trees connected with a single through-silicon-via (TSV). We show that this solution suffers from long wirelength (WL) and high clock power consumption. Our algorithm improves on this solution, minimizes the overall WL and clock power consumption, and provides both pre-bond testability and post-bond operability with minimum skew and constrained slew. Compared with the single-TSV solution, SPICE simulation results show that our multi-TSV approach significantly reduces the clock power by up to 15.9% for two-die and 29.7% for four-die stacks. In addition, the WL is reduced by up to 24.4% and 42.0%.**

***Index Terms*—3-D stacked ICs, clock routing, low-power design, pre-bond test, through-silicon-via (TSV).**

I. Introduction

THREE-DIMENSIONAL system integration has emerged as a key enabling technology to continue the scaling trajectory predicted by Moore's Law for future integrated circuit (IC) generations. With 3-D integration technology, both the average and maximum distance between components can be substantially reduced by placing them on different dies, which translates into significant savings in delay, power, and area. Moreover, it enables the integration of heterogeneous devices, making the entire system more compact and efficient. Nevertheless, the success of 3-D-stacked ICs is predicated on the final post-bond yield, i.e., minimizing the number of good dies bonded to defective dies. Therefore, each die must be tested prior to the bonding process.

Recently, the authors of [2] and [3] demonstrated that there exists a through-silicon-via (TSV) versus wirelength (WL) (and, thus, power) tradeoff in 3-D clock trees: the more TSVs used in the 3-D clock tree, the shorter the total WL. This clearly motivates using more TSVs in a 3-D clock tree. However, it is also shown in [2] and [3] that 3-D clock trees containing multiple TSVs have an interesting property. Only one die in the stack contains a fully connected 2-D clock tree; the other dies contain many small, isolated subtrees. These trees take advantage of TSVs to shorten the total WL, but such a design makes pre-bond testing next to impossible because each clock subtree requires its own probe pad. The state-of-the-art testing equipment, e.g., from [4], has more than ±100 ps overall timing accuracy (OTA). This makes it very challenging to use multiple clock probe pads to provide a low-skew clock signal. In addition, the cost of dedicating so many probes to a single signal is significant. Our paper addresses these issues, providing low-cost methods for designing low-power pre-bond testable clock trees for 3-D-stacked ICs. The contributions of our paper are as follows.

1) We present the first work on pre-bond testable clock routing. Our pre-bond testable clock tree can be used for both pre-bond test and post-bond operation. We introduce two new circuit elements, a *TSV-buffer* and a *redundant tree*, to enable efficient pre-bond testing while minimizing the overall WL and clock power.
2) In order to improve the reliability of our pre-bond testable 3-D clock tree, we develop a slew-aware merging and buffering method to keep the slew rate at each clock sink within a given constraint. This method has the added benefit of reducing the WL and power consumption of the pre-bond testable 3-D clock tree.
3) We show that by allocating the clock source in a middle die in the 3-D stack, our pre-bond testable clock tree will use significantly fewer TSVs while still achieving power and WL savings comparable to other cases.
4) We analyze the impact of the parasitic TSV capacitance on pre-bond testable clock trees in terms of WL, buffer count, and clock power. We demonstrate that a larger capacitance tends to increase the WL and the number of buffers required, in turn increasing the clock power.

Compared with the simple pre-bond testability solution of using a single TSV to connect two complete 2-D trees, our

Manuscript received March 14, 2010; revised July 1, 2010 and September 22, 2010; accepted November 7, 2010. Date of current version April 20, 2011. This research was supported in part by the National Science Foundation, under Career Grants CCF-0546382 and CCF-0811738, in part by the Center for Circuit and System Solutions, and in part by the Interconnect Focus Center. A short version of this manuscript is published in the Proceedings of 2009 IEEE/ACM International Conference on Computer-Aided Design [1]. This paper was recommended by Associate Editor P. Saxena.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: xinzhao@gatech.edu; dean@gatech.edu; leehs@gatech.edu; limsk@ece.gatech.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCAD.2010.2098130

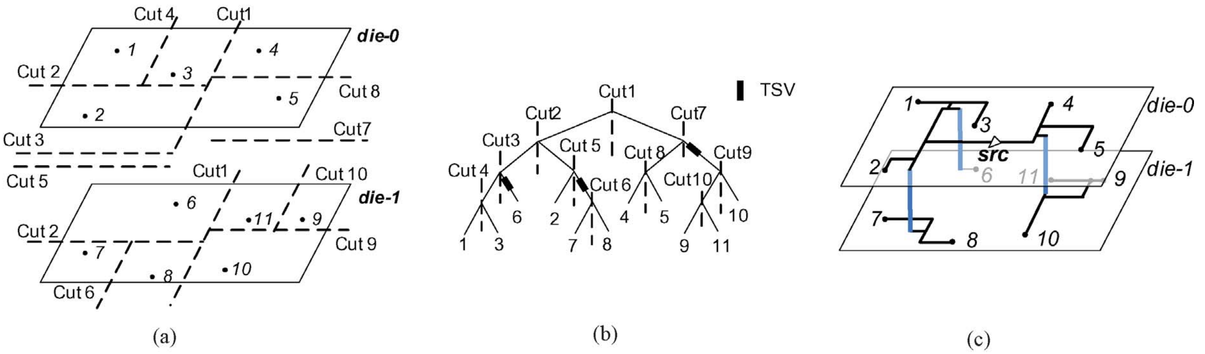


Fig. 1. 3-D clock routing for a two-die stack with a maximum TSV count of three. (a) Top-down partitioning using 3-D-MMM algorithm [2]. (b) Generated abstract binary tree. (c) Final 3-D clock topology.

solution significantly reduces the WL and power consumption in both two-die and four-die 3-D stacks.

The remainder of this paper is organized as follows. Section II presents the summary of related work and its limitation. Section III provides the problem formulation. Section IV presents our pre-bond testable 3-D clock tree routing algorithm. Section V presents our WL and slew-aware buffer insertion algorithm. Experimental results are presented in Section VI, and we conclude in Section VII.

II. Related Work

A. Prior Work on 3-D Clock Routing

The history of clock tree synthesis for 3-D-stacked ICs is short. Pavlidis *et al.* [5] presented measurement data from a fabricated 3-D clock distribution network. Arunachalam and Burleson [6] used a separate layer for the clock distribution network to reduce power. Minz *et al.* [2] presented the first work on 3-D clock routing with the goal of minimizing WL. They also tackled the impact of thermal variations on clock skew. Zhao and Lim [3] presented a comprehensive study of 3-D clock tree synthesis and proposed several design techniques for generating reliable and low-power 3-D clock tree designs. Kim and Kim [7] proposed a clock-embedding method for 3-D clock tree synthesis. They focused on minimizing the TSV count and WL. None of these works address pre-bond testability, unfortunately.

To tackle the 3-D IC testing problem, several testing methods have been investigated. Lee and Chakrabarty [8] presented a comprehensive study of the challenges of testing 3-D ICs. Marinissen and Zorian [9] provided an overview of manufacturing processes in TSV-based 3-D-stacked ICs and discussed the test challenges. To improve testability with stitching WL minimization, 3-D scan chain design approaches are proposed by Wu *et al.* [10]. In [11], Wu *et al.* developed a test-access mechanism (TAM) optimization technique for minimizing the test time of 3-D core-based system-on-chips while constraining the total number of TSVs and the TAM widths. Noia *et al.* [12] addressed the test-wrapper optimization of TSV-based 3-D ICs. The scan-test time is minimized for a core under the constraint of the total number of TSVs available for testing. All these works focus on the post-bond test in 3-D ICs.

Lewis and Lee presented an architectural solution in [13] to the pre-bond testability problem for 3-D die-stacked microprocessors. They discussed how to perform testing for functional modules that are partitioned across multiple dies. They also investigated new design and test methods in [14] to address similar issues for 3-D circuits. Jiang *et al.* [15] presented a heuristic method for optimizing test time and routing cost for both post-bond test and pre-bond wafer-level test. In [16], Jiang *et al.* proposed a layout-driven test-architecture design technique under a constrained pre-bond test pin count.

B. 3-D Abstract Tree Generation

The authors of [2] proposed the 3-D method of means and medians (MMM) algorithm to generate the abstract tree for a set of 3-D clock sinks in a top-down manner. An upper bound for the TSV count (hereafter called *TSV bound*) is a user-defined constraint on the maximum number of TSVs the algorithm can use. The basic idea of 3-D-MMM is to recursively divide the given sink set into two subsets until each sink belongs to its own set. Fig. 1(a) demonstrates the partitioning process based on both x and y coordinates and the TSV bound. At each recursive partitioning step, we divide the given sink set S into two subsets S_A and S_B. The following two cases are considered based on the TSV bound for the current sink set S as follows.

1) If the TSV bound is one, the current sink set needs to be partitioned such that the sinks in the same die belong to the same subset. The connection between S_A and S_B needs one TSV.
2) If the TSV bound is greater than one, the z-dimension coordinates are ignored, and the set is partitioned geometrically by a straight cut line. Since each subset contains sinks from all the dies, multiple TSVs will be needed to connect them.

At the end of partitioning S, the TSV bound for each subset S_A and S_B is determined as follows: 1) estimate the number of TSVs required by each subset, and 2) assign a portion of the TSV bound of S to each subset according to the ratio of the estimated TSV counts. The cut direction is set so as to balance the TSV bound across the subsets. After completing 3-D-MMM partitioning, we obtain an abstract binary tree, as shown in Fig. 1(b). This binary tree represents the hierarchical connection among clock sinks, internal nodes, TSVs, and the clock source.

During the embedding and buffering step, the internal nodes and TSVs in the 3-D abstract tree are placed in a bottom-up fashion, and the buffers are inserted to maintain the zero-skew property (delay is estimated with the Elmore delay model [17]). The classic deferred-merge and embedding (DME) algorithm [18] is extended to generate a topology embedding for the given abstract tree. A cost function that considers the capacitance of the buffers, TSVs, and wires is used for buffer insertion [2].

C. 3-D-MMM Algorithm and Pre-Bond Testing

The 3-D clock tree generated by the 3-D-MMM algorithm poses the following major challenges for pre-bond testing. First, the connected tree on the clock source die is not zero skew pre-bond, as illustrated in Fig. 1(c). The entire 3-D clock tree shown in this figure was constructed for zero-skew operation post-bond. However, without the clock sinks in die-1, the tree in die-0 is missing many of its branches (and thus their parasitic capacitors), which will cause serious timing violations during pre-bond testing that cannot be fixed by slowing down the clock frequency.

Second, each die except the clock source die lacks a complete tree that connects all the sinks in that die, as illustrated in Fig. 1(c). In this figure, there are three subtrees that connect the sinks in die-1. In order to provide a skew-free clock signal during pre-bond test of die-1, we need three probe pads to provide three synchronized clock signals. This cost only increases as more TSVs are used to form more subtrees in order to minimize WL and power consumption.

Thus, the goals of our paper are: 1) to construct a 3-D clock tree that provides a zero-skew clock signal for both pre-bond test and post-bond test and operation; 2) to limit each die to a single clock source (and thus a single probe pad) during pre-bond test; 3) to minimize the WL and silicon costs of the tree; 4) to minimize the overall power consumption; and 5) to bound the clock slew within a given constraint.

III. Problem Formulation and Terminology

The pre-bond testable 3-D clock routing problem is defined as follows: given a set of clock sinks distributed across N dies (where $N > 1$) and a TSV bound, construct a 3-D clock tree such that: 1) during post-bond operation, the tree connects all the sinks with a minimum-skew clock signal, and 2) during pre-bond test, a single 2-D clock tree exists in each die that provides a minimum-skew clock signal to the sinks in that die. The objective is to minimize the WL and clock power given the TSV bound and clock slew bound constraints. The clock sinks may represent flip-flops, clock input pins for IP blocks, or memory blocks. Our pre-bond testable clock routing algorithm can operate under any TSV bound greater than zero, and it constructs a high quality 3-D clock tree in terms of clock skew,[1] WL, power consumption, and clock slew for both pre-bond and post-bond testing and operations.

For an N-die stack clock network, we number the die as die-0, die-1, $\cdots$, die-$(N-1)$ in a top-down order. Given an N-die clock tree design with the clock source located in die-0, the term *post-3-D* refers to the fully connected 3-D clock tree used in post-bond operation, *pre-die-k* is the fully connected 2-D clock tree in die-k for pre-bond test; *sub-die-k* refers to the set of unconnected subtrees in die-k, *red-die-k* refers to the redundant tree used to fully connect a sub-die-k set in a given die. We have the following relations among these trees: 1) *pre-die-k = sub-die-k + red-die-k, when k* > 0, and 2) *post-3-D = pre-die-0 + sub-die-1 + sub-die-2 +* $\cdots$ *+ sub-die-*$(N-1)$.

IV. Pre-Bond Testable Clock Routing

A. Overview

Without loss of generality, we first develop a pre-bond testable clock routing algorithm for a two-die stack. We extend it to the stacks containing more than two dies in Section IV-E. The input to our algorithm includes the location and capacitance of the sinks in each die (die-0 and die-1), a TSV bound (> 0), and a slew constraint. Die-0 is assumed to contain the clock source. Our algorithm consists of two main steps.

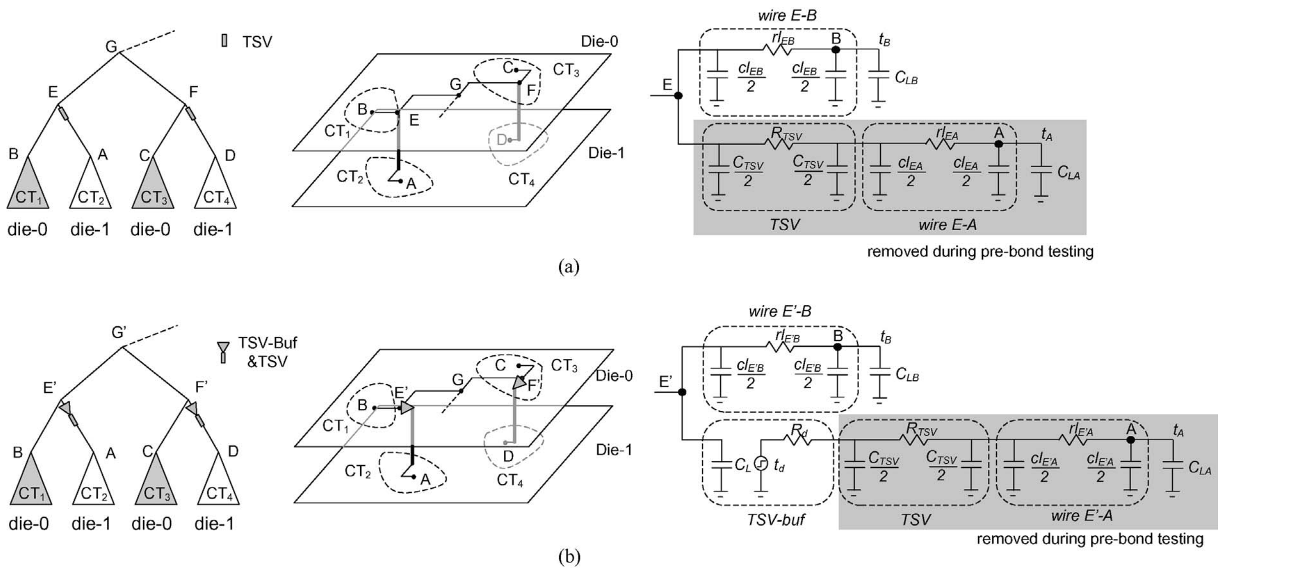
1) *3-D tree construction:* we generate a 3-D clock tree (*post-3-D*) connecting all the sinks in both dies so that:
   a) the overall 3-D tree is zero skew under the Elmore delay model;
   b) the total WL is minimized;
   c) die-0 contains a fully connected 2-D tree (*pre-die-0*) with zero skew. In this case, the 3-D tree is used during post-bond test and operation, while the 2-D tree in die-0 is used for pre-bond test of die-0. We utilize so called "TSV-buffers" to ensure that the 2-D tree in die-0 maintains zero skew in *both* pre-bond and post-bond configurations.
2) *Redundant tree routing:* if multiple TSVs are used, the 3-D tree construction step generates a 3-D tree, where die-1 contains several separate subtrees (*sub-die-1*). In this case, we route a so-called "redundant tree" in die-1 (*red-die-1*) to connect the roots of the subtrees in die-1 and form a single fully connected 2-D tree (*pre-die-1*) with:
   a) an estimated zero skew;
   b) a minimum total WL.

   This 2-D tree is used for the pre-bond test of die-1. Transmission gates (*TG*s) are inserted to disconnect the redundant tree for post-bond operation.

B. TSV-Buffer Insertion

Testing die-0 pre-bond requires a fully connected clock tree in die-0 so that the clock signal is delivered to all die-0 sinks using a single test probe. As mentioned earlier, if multiple TSVs are used, the 3-D tree construction step gives a 3-D tree, where die-0 contains a single fully connected tree and die-1 contains a forest of small subtrees. During pre-bond test, the two dies are separate and tested individually. In this case, the 2-D tree in die-0 can be used without any additional

[1] In the pre-bond testable clock routing, our algorithm generates zero-skew clock trees based on the Elmore delay model [17]. To obtain accurate clock-related metrics, we then extract the netlist, and report the SPICE simulation results, including delay, skew, slew, and power consumption.



(a)

(b)

Fig. 2. (a) 3-D clock tree built with TSVs, where the separation of die-0 and die-1 skews the tree in die-0. (b) 3-D clock tree built with TSV-buffers, where the separation of die does not skew the die-0 tree.

modification. However, the skew of this tree may no longer be zero because the downstream capacitances of the subtrees in die-1 are not present. This additional skew will either slow down or corrupt the testing process.

To avoid this high-skew situation, we employ our TSV-buffer, simply a buffer inserted right before a TSV. In our test-aware DME (*TaDME*) algorithm, we add a TSV-buffer for each TSV and route the tree accordingly under the zero-skew constraint. In this case, the TSV-buffers are inserted in die-0, where the clock source is located. Since the buffers shield die-0 from the downstream capacitance, die-0 remains zero-skew when tested pre-bond. The outcome of TaDME is a zero-skew 3-D tree that contains a zero-skew 2-D tree in die-0 for pre-bond test.

In what follows, we describe how our TaDME algorithm modifies the traditional DME algorithm to construct a zero-skew 3-D clock tree in the presence of TSV-buffers. A key step in TaDME is bottom-up recursive tree merging. Given a pair of zero-skew subtrees that must be merged, our goal is to determine the merging segment (the set of potential locations for the merging points) and to connect it to the root nodes of the subtrees so that the new merged tree is also zero-skew. Fig. 2(a) shows the traditional merging process as used in the original DME algorithm, where the merging segment of internal node E is determined based on the parasitics of the TSVs, wires, downstream capacitances, and internal delays of the two subtrees. In this case, if the right branch (TSV, edge (E, A), and CT_2) of the overall tree is missing, the delay from E to B will change due to the change in the downstream capacitance at node E. However, if we use a TSV-buffer as shown in Fig. 2(b), the delay from E' to B will not change even if we remove the right branch. This is because the TSV-buffer hides the downstream capacitance at node E'.

The following notations are used in Fig. 2. r and c denote the unit-length wire resistance and capacitance, respectively. R_d is the output resistance of a buffer, C_L is the input capacitance of a buffer, and t_d is the intrinsic delay of a buffer. R_{TSV} and C_{TSV} are the resistance and capacitance of a TSV. Die-0 contains a subtree CT_1 with the root B and a loading capacitance C_{LB}. The internal delay from B to the sinks of CT_1 is t_B. Similar symbols are used for CT_2. A clock wire of length l is modeled as a π-type circuit with a resistor (rl) and two capacitors ($cl/2$). We also model the TSVs with π-type circuits with resistance R_{TSV} and two capacitances $C_{TSV}/2$. Note that the downstream capacitance at the internal node E' in Fig. 2(b) is $cl_{E'B} + C_{LB} + C_L$ *both* before and after the dies are bonded. Thus, TSV-buffers allow us to build a 3-D tree for die-0 that is zero-skew both pre-bond and post-bond.

In the bottom-up merging process, we require that the delay from E' to sinks in CT_1 (through $B = d_{E', CT_1}$) be equal to delay to the sinks of CT_2 (through $A = d_{E',CT_2}$), that is

$$d_{E', CT_1} = d_{E', CT_2}. \tag{1}$$

Referring to the merging structure in Fig. 2(b), d_{E', CT_1} and d_{E', CT_2} can be expressed as follows:

$$\begin{aligned} d_{E', CT_1} &= rl_{E'B}(cl_{E'B}/2 + C_{LB}) + t_B & (2)\\ d_{E', CT_2} &= t_d + R_d(C_{TSV} + cl_{E'A} + C_{LA}) + \\ &\quad R_{TSV}(C_{TSV}/2 + cl_{E'A} + C_{LA}) + \\ &\quad rl_{E'A}(cl_{E'A}/2 + C_{LA}) + t_A & (3) \end{aligned}$$

where t_A is the internal delay from A to sinks of CT_2, and C_{LA} is the downstream capacitance of node A. If there is no detour and given the distances between E' and A ($l_{E'A}$) and between E' and B ($l_{E'B}$), it follows that

$$l_{E'B} + l_{E'A} = L \tag{4}$$

where L is the minimum merging distance between A and B. $l_{E'A}$ and $l_{E'B}$ can be determined by solving (1)–(4).

If $l_{E'A}$ or $l_{E'B}$ is negative, a wire detour is required. For example, when $l_{E'A}$ is negative, $l_{E'B}$ must be longer than L to obtain a zero-skew merging. In this case, $l_{E'A}$ is set to zero,

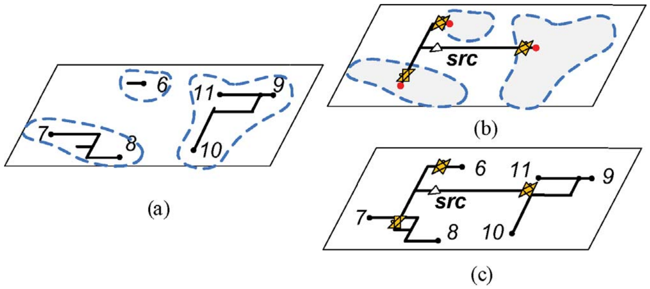


Fig. 3. Redundant tree insertion in die-1. (a) Extract sinks from subtrees. (b) Generate a redundant tree and insert TGs. (c) Final pre-bond testable clock tree in die-1. The extra control signal that connects the TGs is not shown here for simplicity.

and $l_{E'B}$ is calculated by solving the (1)–(3). If the calculated $l_{E'B}$ is too long, we insert a clock buffer along the edge $E'B$. Equation (2) is updated correspondingly. The decision to avoid a detour with a buffer is made by a cost function that considers the capacitance of clock wires, buffers, and TSVs; we use a wire detour if the cost is less than that of buffer insertion and satisfies the slew constraint.

C. Redundant Tree Insertion

Pre-bond test of die-1 requires a fully connected clock tree so that the clock signal is delivered to all the sinks in die-1 from just a single test probe. As mentioned earlier, when multiple TSVs are used for WL reduction, the 3-D tree construction generates a forest of subtrees in die-1. Therefore, our goal is to combine these subtrees into a single fully connected clock tree with zero clock skew and minimum overall WL. We accomplish this by adding a redundant tree that connects the roots of the subtrees while maintaining zero skew. We use this fully connected tree during the pre-bond test of die-1. Note that the redundant tree is not used during post-bond test and operation. We use TGs to disconnect the redundant tree.

The redundant tree routing is done using a conventional algorithm: 1) construct a binary abstract tree in a top-down fashion; 2) insert a TG at each sink node; and 3) embed and buffer the abstract tree under the zero-skew and minimal WL goals. Fig. 3 shows a sample flow. Given many subtrees in die-1, we first extract a new set of sinks based on the subtrees, as in Fig. 3(a). Then, we construct a 2-D clock tree for this extracted set, as in Fig. 3(b). Fig. 3(c) shows the final pre-bond testable clock tree in die-1 (*pre-die-1*), which consists of three subtrees (*sub-die-1*) and one redundant tree (*red-die-1*). Last, we connect the enable input of the TGs using an extra control wire. In order to minimize the routing overhead, we need to minimize the total WL of this control signal. We use the rectilinear minimum spanning tree (RMST) algorithm-pack [19] for this purpose. The cost of this overhead is reported in Section VI-C.

D. Putting It Together

Upon the completion of our algorithm, we obtain fully connected zero-skew 2-D clock trees for both die-0 and die-1 as well as a fully connected zero-skew 3-D tree for the entire

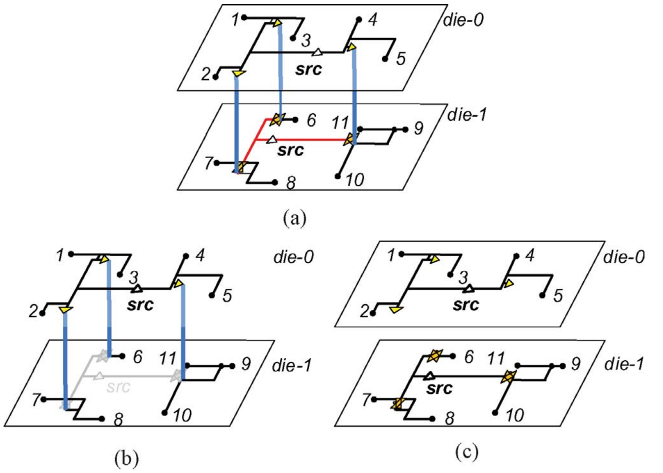


Fig. 4. Example of the post-bond operations and pre-bond test using our 3-D clock tree. (a) Pre-bond testable 3-D clock tree. (b) *Post-3-D* in post-bond operation with TGs turned off. (c) *Pre-die-0* and *pre-die-1* in pre-bond test with TGs turned on.

stack. In die-1, we turn on the TGs to connect the redundant tree to the subtrees for pre-bond test. Once the pre-bond testing is complete, we turn off the TGs to disconnect the redundant tree. By doing this, the original zero-skew 3-D tree is used for post-bond test and normal operation. We will show in our experimental results section that our 3-D trees with multiple TSVs, TSV-buffers, and TGs plus the control signal consume significantly less power than a simple single-TSV solution.

Fig. 4(a) shows an illustration of the entire design flow. In post-bond operation, the TGs are turned off and the *pre-die-0* and *sub-die-1* trees are connected with TSVs to form the *post-3-D* tree, as shown in Fig. 4(b). In pre-bond test, the *pre-die-0* tree can be reused with zero skew to test die-0, as shown in Fig. 4(c). To test die-1, we turn on the TGs, and the *red-die-1* and *sub-die-1* trees form the zero-skew *pre-die-1* tree, as shown in Fig. 4(c).

E. Multiple-Die Extension

For a stack with more than two dies, we face the same challenges of creating clock trees for pre-bond test. We take a four-die stacked clock tree in Fig. 5 as an example. The clock source is located in die-0. If we apply the 3-D-MMM algorithm [2], the resulting *post-3-D* tree contains the following topology: 1) die-0 has a complete clock tree connecting all the sinks in die-0, and 2) the non-source dies (die-1, die-2, and die-3) have each a *sub-die-k* (k = 1, 2, 3), which is connected to the clock source through ten TSVs.

Our pre-bond testable clock routing algorithm for a two-die stack can be easily extended to larger die stacks with an arbitrary clock source location. Our basic 3-D tree construction algorithm presented in Section II-B generates a 3-D tree, where die-s (defined as containing the clock source; die-0 in Fig. 5) has a single, fully connected tree, while all the other dies have a forest. Our TSV-buffer insertion algorithm is extended as follows. During the bottom-up merging process:

1) if a TSV connects die-s and a non-source die-k where (k $\neq$ s), we insert a TSV-buffer in die-s;

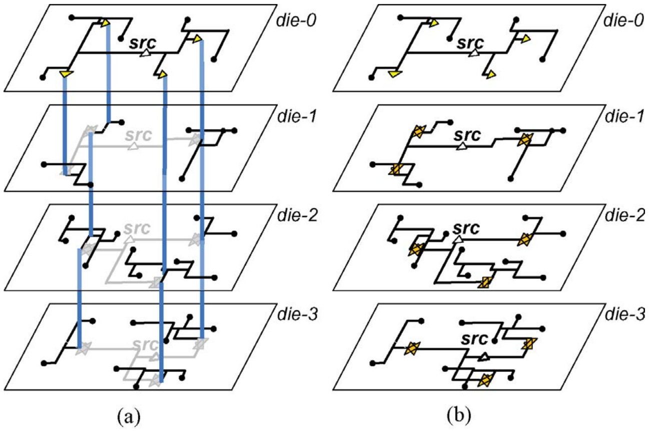


Fig. 5. Example of a pre-bond testable clock routing in a four-die stack. (a) 3-D clock tree, in post-bond. (b) 2-D clock trees, in pre-bond test.

2) if a TSV connects non-adjacent dies and passes through die-s [e.g., connecting die-(s−1) and die-(s+1)], we insert a single TSV-buffer in die-s;
3) if a TSV does not connect to or travel through die-s, no TSV-buffer is required.

Once the TSV-buffer insertion and embedding and buffering are completed, we add redundant trees to the non-source dies. In addition, we insert TGs at the root of each subtree and add a global control signal to connect all the TG enable inputs in each die. This allows us to use the redundant trees for pre-bond test (TGs on) and disable them during post-bond test and operation (TGs off). The outcome of the whole process is: 1) a single zero-skew 3-D clock tree for post-bond test and normal operation; 2) a zero-skew 2-D clock tree in each die for pre-bond test; and 3) a global control signal that connects the enable inputs of the TGs in each die. Fig. 5 shows an illustration of the pre-bond testable and post-bond operational 3-D clock tree for a four-die stack.

V. Buffering for Wirelength and Slew Control

This section presents our buffering strategies for balancing WL and controlling slew.

A. Wirelength Balancing with Clock Buffers

Our pre-bond testable 3-D clock routing algorithm inserts two kinds of buffers: clock buffers and TSV-buffers. Clock buffers, as discussed in Section II-B, are mainly used to control delay and skew. These clock buffers are usually inserted close to the clock source and drive large loads to reduce the delay along the clock paths. The TSV-buffers, as discussed in Section IV-B, are inserted at every TSV location in the clock source die to ensure that the clock tree in that die is also zero-skew during pre-bond test.

Our observations indicate that TSV-buffers may unbalance the WL during the bottom-up merging process. Consider the example of two subtrees CT_1 and CT_2 in die-0 and die-1, respectively; we must use a TSV-buffer in die-0 to merge these subtrees. As shown in Fig. 2(b), TSV-buffer insertion can increase the delay from E' to CT_2. If the internal delay of CT_2 is already much greater than that of CT_1, adding the TSV-buffer only makes the difference worse. If the difference is too large, wire snaking is required to balance the delays and to achieve a zero-skew merged tree. Thus, the addition of a TSV-buffer has led to a significant clock WL overhead in die-0.

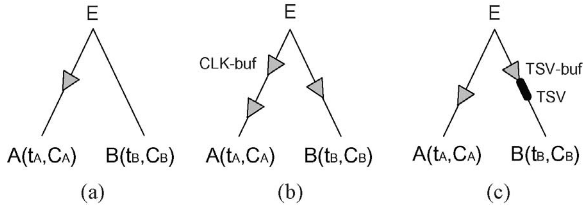


Fig. 6. Examples of clock buffer and TSV-buffer insertion. (a) Clock buffer is inserted to balance the delay of the two branches, where $t_A < t_B$. (b) Multiple clock buffers are inserted if the wires are long and/or the download capacitance is large. (c) Clock buffer is inserted along with a TSV-buffer to balance the delay.

To mitigate this overhead, we add extra clock buffers to die-0 to balance the internal delays and eliminate snaking. Specifically, when a TSV-buffer significantly unbalances the delay, we insert an extra clock buffer on the other branch as a counter-balance. In Fig. 2(b), we add an extra clock buffer along $E' - B$. We observe that this delay balancing scheme reduces the overall WL in die-0. We also observe that few clock buffers are required in this way because such unbalances do not occur frequently.

B. Slew Rate Control with Clock Buffers

Clock slew rate control is an important reliability issue for high-speed clocking. If the slew rate is too low, i.e., if it takes too long for the clock signal to rise or fall, setup and hold times may be violated, a problem which cannot be fixed with a lower clock frequency. Existing work on slew-aware clock tree synthesis relies on buffer insertion [20]–[23]. Buffers are added along the clock paths so that the output load of each buffer is limited. This bounding condition, denoted as *cmax* in the literature, is shown to be effective in controlling the slew rate; a smaller *cmax* value improves the slew rate but requires that more buffers are inserted. Most existing works insert buffers in a given clock tree as a post-processing step to improve the slew rate under various constraints: buffer area, clock power, and others. This post-synthesis slew-aware buffer insertion must be done carefully to avoid introducing new clock skew. This may constrain the location of the buffers.

Our strategy is to tackle the slew rate issue *during* the construction of the pre-bond testable clock trees by adding buffers to meet the *cmax* constraint. Specifically, we insert clock buffers, together with TSV-buffers, during the bottom-up merging process so that *cmax* is satisfied for both types of buffers. We add clock buffers along the paths from the merging node to the subtree root nodes if the downstream capacitance at the merging node exceeds *cmax*. Depending on the load, we may insert multiple clock buffers to meet the *cmax* requirement.

Fig. 6 shows several possible scenarios for clock buffer and TSV-buffer insertion. In summary, our clock tree synthesis algorithm uses three criteria for buffer insertion during the bottom-up merging process.

1) *For pre-bond testability:* we add a TSV-buffer for every TSV connecting to the clock source die.
2) *For WL reduction:* we add a clock buffer to correct unbalances in the delays of two merging subtrees as discussed in the previous section.
3) *For slew rate control:* we add clock buffers if the downstream capacitance of any buffer exceeds the given limit *cmax*.

VI. Experimental Results

We implemented our algorithm using C++/STL on Linux. We use five benchmarks from the IBM suite [24] and four from the ISPD clock network synthesis contest suite [25]. Since these designs are for 2-D ICs, we obtain 3-D designs by randomly partitioning the clock sinks across the multiple dies and scaling the footprint area by $\sqrt{2}$ and $\sqrt{4}$ for two-die and four-die stacks, respectively.

We use technology parameters from the 45 nm predictive technology model [26]; the unit-length wire resistance is $0.1\,\Omega/\mu$m, and the unit-length wire capacitance is $0.2\,$fF$/\mu$m. The sink capacitance values range from 5 fF to 80 fF. The buffer parameters are $R_d = 122\,\Omega$, $C_L = 24\,$fF, and $t_d = 17\,$ps. We use $10\,\mu$m$\times 10\,\mu$m via-last TSVs with $20\,\mu$m height and $0.1\,\mu$m liner oxide thickness. By simulating the TSV structure with Synopsys Raphael [27], we determine the TSV parasitics to be $R_{TSV} = 0.035\,\Omega$ and $C_{TSV} = 15.48\,$fF. The clock frequency is set to 1 GHz and the supply voltage (V_{dd}) to 1.2-V.[2] The maximum load capacitance for each buffer *cmax* is 300 fF for slew rate control.

In SPICE simulation, wire segments and TSVs are represented as π models, and clock buffers and TSV-buffers are represented as inverter pairs. The simulated clock skew and slew tolerances are 3% and 10% of the clock period, respectively. We report WL in μm, clock power in mW, skew and slew in ps, and capacitance in fF.

A. TSV-Buffer and TG Model Validation

In pre-bond testable clock routing, we utilize TSV-buffers and TGs to facilitate pre-bond test and post-bond test and operation. Fig. 7 shows the equivalent circuits used for SPICE validation of the TSV-buffers and TGs. We simulate a post-bond 3-D clock tree in a two-die stack and two pre-bond testable 2-D clock trees in die-0 and die-1. Node A is the clock source for post-bond operation. Sink C in die-0 and sink E in die-1 have loading capacitances of C_{LC} and C_{LE}, respectively. Node B and D are connected by a TSV-buffer and a TSV. The edge (D, E) is a subtree in die-1 and is connected to F, the clock source for pre-bond test of die-1, via a TG. C_{LC} and C_{LE} are set to 5 fF. Wire (A, B), (B, C), (D, E), and (F, D) all have $500\,\mu$m length.

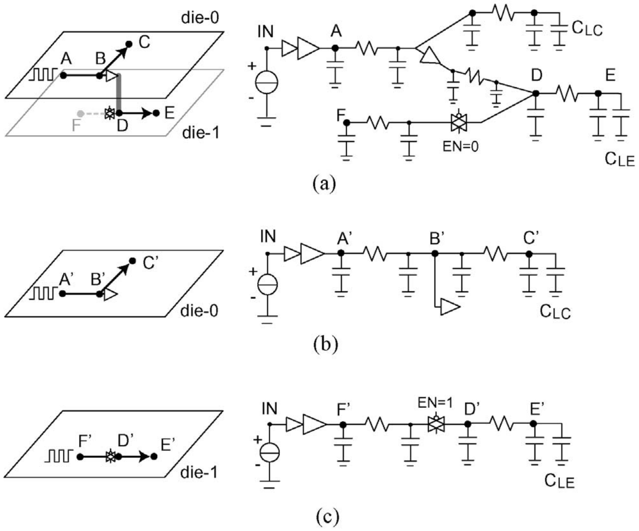


Fig. 7. Circuit models. (a) For the post-bond 3-D clock tree. (b) For the pre-bond testable 2-D clock tree in die-0. (c) For the pre-bond testable 2-D clock tree in die-1.

First, we observe from SPICE simulation that the delay from A to C in Fig. 7(a) is 42.21 ps, which is the same as that from A' to C' in Fig. 7(b). This verifies that die-0 is zero skew before die-1 is attached and so the TSV-buffer has done its job. Second, the TG has 14.2 fF capacitance between node D and the ground when it is off. This TG completely blocks the clock signal from A to F. When the TG is on for pre-bond testing on die-1, however, it has $108\,\Omega$ between its input and output nodes, 16.4 fF between its input and the ground, and 18.4 fF between its output and the ground. The intrinsic delay of a TG is 1.04 ps. Under this model, the calculated delay from F' to E' is 54.13 ps, which closely matches the simulated delay of 54.14 ps.

B. Sample Trees and Waveforms

Fig. 8 shows a series of pre-bond testable clock trees for the circuit r_1 from the IBM suite given a TSV bound of 10. The TSVs are shown as black dots and the clock sources as triangles. Fig. 8(a) is the zero-skew 3-D clock tree for post-bond test and normal operation. This 3-D clock tree contains ten TSVs. The solid and dotted lines represent the clock trees in die-0 and die-1, respectively. Note that die-1 contains many subtrees (dotted lines) that are not connected to each other except through die-0. Fig. 8(b) shows the zero-skew pre-bond testable 2-D clock tree for die-0, which is identical to the solid line clock tree in Fig. 8(a). Fig. 8(c) shows the zero-skew pre-bond testable 2-D clock tree for die-1, which contains all the subtrees (dotted lines) in die-1 and the redundant tree (solid line) which connects them.

Fig. 9 shows two groups of clock waveforms for benchmark r_5, where each group contains 25 waveforms (one each for the 25 sinks in each tree). The first group (shown on top) is from the post-bond 3-D clock tree, the second group (shown on bottom) is from the pre-bond testable 2-D clock tree for die-0. We first observe that the 25 waveforms are almost identical, which is desirable. In addition, the two groups have

[2] Note that our clock trees with single and multiple TSVs are simulated under the same V_{dd}, and the power savings mainly come from the capacitance reduction. Therefore, the efficiency of our algorithm in low power and pre-bond testability apply on different V_{dd} (e.g., from 1.2-V to 1.0-V).

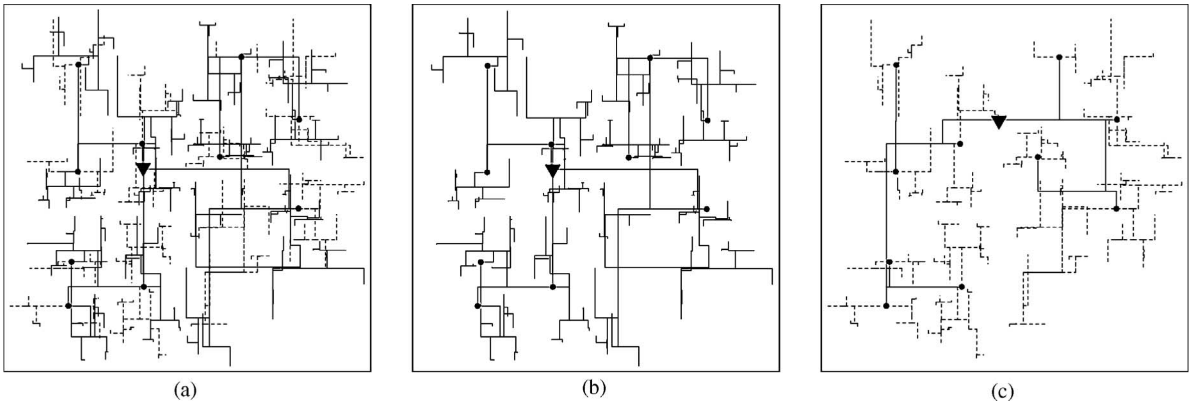


Fig. 8. Pre-bond testable clock trees for circuit $r1$ in a two-die stack for a TSV bound of 10. The TSVs and the clock sources are represented by black dots and triangles, respectively. (a) Post-bond 3-D clock tree, where the solid and dotted lines denote the trees in die-0 and die-1, respectively. (b) Pre-bond testable 2-D clock tree for die-0. (c) Pre-bond testable 2-D clock tree for die-1, where the redundant tree and the subtrees are drawn in solid and in dotted lines, respectively.

TABLE I

WIRELENGTH, CLOCK POWER, AND SKEW RESULTS FOR POST-BOND TESTABLE 3-D CLOCK TREES AND PRE-BOND TESTABLE 2-D CLOCK TREES

			Post-Bond 3-D			Pre-Bond Testable Die-0			Pre-Bond Testable Die-1					
ckt	#Sinks	#TSVs	WL	Power	Skew	WL	Power	Skew	WL	WL-sub	WL-red	WL-TG	Power	Skew
r_1	267	57	227 141	128.4	13.7	166 691	103.0	13.5	150 219	60 450	89 769	62 732	68.2	13.0
r_2	598	95	488 987	274.1	14.2	328 914	196.0	14.1	302 023	160 073	141 950	109 031	148.6	11.8
r_3	862	183	616 077	361.6	15.5	444 156	280.5	15.5	429 950	171 921	258 029	161 561	201.9	16.2
r_4	1903	265	1 311 290	763.2	15.5	889 460	536.4	14.9	846 980	421 830	425 151	259 442	422.1	15.1
r_5	3101	269	1 998 950	1115.0	29.1	1 255 760	715.9	29.1	1 236 417	743 190	493 227	310 855	615.9	20.9
$ispd09f11$	121	44	129 391	73.3	9.4	99 393	64.1	9.2	99 169	29 998	69 171	51 214	44.3	6.3
$ispd09f12$	117	36	127 763	71.2	6.8	96 093	60.4	6.2	93 625	31 669	61 956	42 134	42.0	5.7
$ispd09f21$	117	42	136 676	75.6	5.0	107 834	67.0	4.7	101 968	28 841	73 127	52 241	45.0	7.3
$ispd09f22$	91	30	80 977	46.8	15.3	61 504	40.4	15.2	59 870	19 473	40 397	29 449	26.4	14.9
Ratio			1.00	1.00	1.00	0.72	0.79	0.97	0.69	0.28	0.41	0.29	0.57	0.94

similar waveforms, which demonstrates that the TSV-buffer does maintain the balance of the tree in both pre-bond and post-bond test configurations. Second, the SPICE simulation shows that the clock skew among all sinks in both cases is 29.1 ps, observed by the width of waveforms at 50% V_{dd}. Third, the maximum slew rate is 88.4 ps, measured as the rise time from 10% to 90% of V_{dd} (or fall time from 90% to 10% V_{dd}) at the slowest node. Both the skew and slew values are within the tolerances (3% and 10% of the clock period, respectively).

C. Wirelength, Skew, and Power Results

Table I shows the WL (μm), power consumption (mW), and skew (ps) results for the post-bond 3-D clock tree (*post-3-D*), the pre-bond testable 2-D clock tree for die-0 (*pre-die-0*) and die-1 (*pre-die-1*). For die-1, we report the total wirelength (*WL*), and the WL of the subtrees (*WL-sub*), redundant tree (*WL-red*), and TG control signal (*WL-TG*). In this case, the WL of the pre-bond testable clock tree for die-1 is equal to the sum of *WL-sub* and *WL-red*. In addition, the WL of the post-bond 3-D clock tree is the sum of the WL of *pre-die-0* and *WL-sub* from *pre-die-1*.

Based on the WL-related columns, we observe that: 1) the total WL of *pre-die-0* and *pre-die-1* are comparable (0.72 versus 0.69 in ratio); 2) in several cases, the WL of the redundant tree is about 2× of the total WL of the subtrees in die-1 (0.41 versus 0.28); and 3) in several cases, the WL of the TG control signal is about half of the redundant tree in die-1 (0.29 versus 0.41).

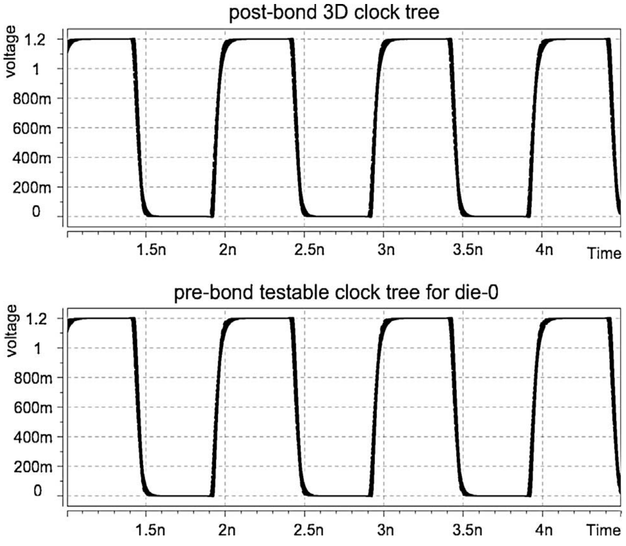


Fig. 9. Clock waveforms from the post-bond 3-D clock tree and the pre-bond testable 2-D clock tree for die-0. We superimpose the waveforms of the 25 clock sinks in r_5. Clock frequency is 1 GHz, skew is 29.1 ps, and maximum slew rate is 88.4 ps.

The total clock routing resource cost is equal to the sum of *post-3-D* and *WL-red* from *pre-die-1*. Normalizing to the WL

TABLE II
COMPARISON BETWEEN SINGLE-TSV AND MULTI-TSV DESIGNS

			Single TSV				Multi-TSV						
												Reduction (%)	
	ckt	#Sinks	#Bufs	WL	Power	Skew	#TSVs	#Bufs	WL	Power	Skew	WL	Power
Two-die	r_1	267	327	279 796	145.0	12.7	57	324	227 141	128.4	13.7	18.8	11.4
	r_2	598	693	600 880	310.6	12.5	95	684	488 987	274.1	14.2	18.6	11.8
	r_3	862	928	765 397	404.3	16.1	183	925	616 077	361.6	15.5	19.5	10.6
	r_4	1903	1982	1 576 510	848.7	15.3	265	1963	1 311 290	763.2	15.5	16.8	10.1
	r_5	3101	2528	2 344 960	1242.0	22.2	269	2449	1 998 950	1115.0	29.1	14.8	10.2
	*ispd*09*f*11	121	212	168 500	85.4	7.6	44	201	129 391	73.3	9.4	23.2	14.1
	*ispd*09*f*12	117	215	164 966	84.2	5.8	36	193	127 763	71.2	6.8	22.6	15.5
	*ispd*09*f*21	117	226	180 867	89.9	9.4	42	211	136 676	75.6	5.0	24.4	15.9
	*ispd*09*f*22	91	106	106 401	53.2	15.1	30	111	80 977	46.8	15.3	23.9	12.1
Four-die	r_1	267	318	272 355	141.8	10.5	248	325	160 394	111.4	13.3	41.1	21.4
	r_2	598	700	582 115	304.5	14.4	434	647	353 646	233.9	15.7	39.2	23.2
	r_3	862	945	735 299	398.0	14.9	718	922	442 903	317.1	13.7	39.8	20.3
	r_4	1903	1956	1 532 220	831.1	14.8	1651	2011	908 375	675.6	16.5	40.7	18.7
	r_5	3101	2939	2 312 930	1272.0	22.2	2469	3134	1 368 370	1041.0	20.3	40.8	18.2
	*ispd*09*f*11	121	216	159 752	83.1	8.4	129	176	93 440	60.0	5.8	41.5	27.8
	*ispd*09*f*12	117	208	155 542	80.9	8.9	114	160	90 281	56.8	10.2	42.0	29.7
	*ispd*09*f*21	117	212	163 816	83.0	17.8	102	160	99 179	58.4	7.8	39.5	29.6
	*ispd*09*f*22	91	99	98 123	48.7	18.0	81	88	57 342	36.1	14.7	41.6	25.9

TABLE III
BUFFER USAGE BETWEEN THE SINGLE AND MULTI-TSV CASES

	Single TSV			Multi-TSV			
ckt	#Bufs	#TBs	#CBs	#TSVs	#Bufs	#TBs	#CBs
*r*1	327	1	326	57	324	57	267
*r*2	693	1	692	95	684	95	589
*r*3	928	1	927	183	925	183	742
*r*4	1982	1	1981	265	1963	265	1698
*r*5	2528	1	2527	269	2449	269	2180
*ispd*09*f*11	212	1	211	44	201	44	157
*ispd*09*f*12	215	1	214	36	193	36	157
*ispd*09*f*21	226	1	225	42	211	42	169
*ispd*09*f*22	106	1	105	30	111	30	81

We report the total number of buffers (#Bufs), TSV-buffers (#TBs), and clock buffers (#CBs). The number of dies is 2.

TABLE IV
IMPACT OF TSV-BUFFER INSERTION

		no	yes	% Increase	
ckt	#TSVs	Skew	Skew	WL	Power
*r*1	248	47.3	9.6	−0.77	7.95
*r*2	434	34.8	13.6	3.89	6.27
*r*3	718	38.5	12.3	3.11	9.19
*r*4	1651	45.2	14.9	2.72	11.28
*r*5	2469	48.0	15.9	4.56	11.17
*ispd*09*f*11	129	30.4	3.9	−1.69	8.28
*ispd*09*f*12	114	33.2	6.8	0.06	7.47
*ispd*09*f*21	102	24.7	4.4	−1.66	3.34
*ispd*09*f*22	81	33.3	13.6	−2.68	2.30

"% Increase" refers to the WL and power consumption increases from TSV-buffer insertion.

of *post-3-D*, the overall WL of the pre-bond testable clock tree and its redundant trees is 1.41. We can derive the following: die-0 and die-1 utilize 51% and 49% of the total clock routing resource, respectively. In the post-bond operations, the *post-3-D* consumes 71% of the clock routing resource. This means that 29% of the clock resource is used for the pre-bond test only. Note that the redundant tree and the TG control signal are used only during the pre-bond testing for die-1. This non-negligible overhead is compensated by the significant power savings to be discussed in Section VI-D.

Last, the clock skew values do not exceed 30 ps, satisfying our 3% of the clock period constraint on the simulated skew. Die-0 consumes more clock power than die-1, primarily due to the TSV-buffers inserted in die-0.

D. Comparison with the Single-TSV Approach

Our baseline 3-D clock tree contains a single, fully connected zero-skew clock tree in each die; these trees are connected with a single TSV in the two-die stacks and a single column of TSVs in taller stacks. Table II compares the WL (μm), clock power (mW), and skew (ps) results from the SPICE simulation. In the multi-TSV designs, we choose the TSV count that gives us the minimum power by an exhaustive search, wherein we sweep the TSV bound from 2 to infinity, construct a 3-D clock tree for each bound, and simulate the power consumption. The clock synthesis time for each tree is less than 1 s in all cases.

We make the following observations. First, our multi-TSV approach significantly outperforms the single-TSV approach in terms of WL: 14.8% to 24.4% reductions for the two-die stacks and 39.2% to 42.0% reductions for the four-die stacks. Similarly, power savings for the clock trees are 10.1–15.9% for the two-die cases and 18.2–29.7% for the four-die cases. These results convincingly demonstrate the benefits of our multi-TSV approach. Second, the total number of buffers (#Bufs) used in the clock trees consists of the clock buffers and the TSV-buffers. Table III shows the detailed buffer usages in the two-die cases, including the total number of buffers (#Bufs), the TSV-buffer count (#TBs), and the clock buffer count (#CBs). We observe that a similar number of buffers is used in both the single and the multi-TSV trees. In the single-TSV design, buffers are inserted to control the WL and slew in each die. In the multi-TSV policy, we need more TSV-buffers to ensure pre-bond testability but use fewer clock buffers. This is because the total WL is shorter in the multi-TSV designs and the TSV-buffers have positive impact on slew control.

TABLE V
IMPACT OF THE CLOCK SOURCE LOCATION IN FOUR-DIE 3-D STACKS

	src in the Top Die (Die-0)					src in the Middle Die (Die-1)							
											Reduction (%)		
ckt	#TSVs	#Bufs	WL	Power	Skew	#TSVs	#Bufs	WL	Power	Skew	#TSVs	Power	WL
*r*1	248	325	160 394	111.4	13.3	208	300	163 249	110.0	12.1	16.1	1.3	−1.8
*r*2	434	647	353 646	233.9	15.7	394	631	352 561	232.6	19.7	9.2	0.6	0.3
*r*3	718	922	442 903	317.1	13.7	620	891	435 177	312.3	13.9	13.6	1.5	1.7
*r*4	1651	2011	908 375	675.6	16.5	1449	1976	893 178	667.9	20.8	12.2	1.1	1.7
*r*5	2469	3134	1 368 370	1041.0	20.3	2208	3022	1 349 334	1027.0	22.4	10.6	1.3	1.4
*ispd*09*f*11	129	176	93 440	60.0	5.8	103	164	94 034	59.3	9.3	20.2	1.2	−0.6
*ispd*09*f*12	114	160	90 281	56.8	10.2	97	163	88 850	57.0	7.3	14.9	−0.4	1.6
*ispd*09*f*21	102	160	99 179	58.4	7.8	94	162	95 920	58.1	7.5	7.8	0.6	3.3
*ispd*09*f*22	81	88	57 342	36.1	14.7	71	85	59 417	37.1	19.7	12.3	−2.7	−3.6

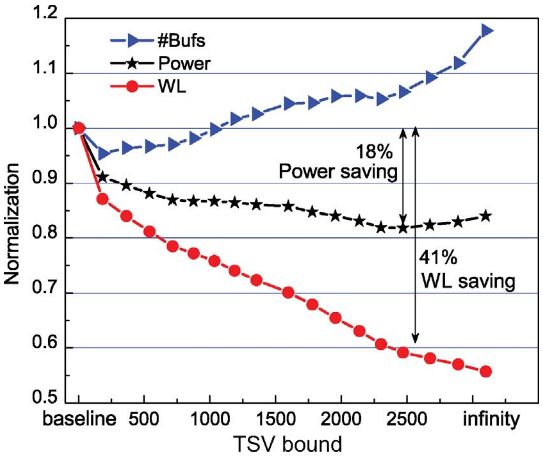


Fig. 10. Impact of the TSV bound constraint on WL, buffer count, and clock power consumption based on the four-die stack of r_5. The baseline is the single-TSV approach.

TABLE VI
STACKED-TSV DISTRIBUTION IN FOUR-DIE 3-D STACKS

	Clock src in the Top Die				Clock src in the Middle Die		
		Stacked TSV Distribution				Stacked TSV Distribution	
ckt	#TSVs	1-stack	2-stack	3-stack	#TSVs	1-stack	2-stack
*r*1	248	1 × 106	2 × 41	3 × 20	208	1 × 144	2 × 32
*r*2	434	1 × 239	2 × 69	3 × 19	394	1 × 282	2 × 56
*r*3	718	1 × 303	2 × 137	3 × 47	620	1 × 406	2 × 107
*r*4	1651	1 × 665	2 × 307	3 × 124	1449	1 × 901	2 × 274
*r*5	2469	1 × 1125	2 × 444	3 × 152	2208	1 × 1464	2 × 372
*f*11	129	1 × 40	2 × 28	3 × 11	103	1 × 67	2 × 18
*f*12	114	1 × 41	2 × 29	3 × 5	97	1 × 65	2 × 16
*f*21	102	1 × 39	2 × 24	3 × 5	94	1 × 58	2 × 18
*f*22	81	1 × 27	2 × 15	3 × 8	71	1 × 43	2 × 14
Ratio (%)	100	40.5	39.4	20.1	100	65.5	34.5

The clock source is located in the top die (die-0) or the middle die (die-1). Note that we do not need 3-stack TSVs for the middle-die case.

E. Impact of TSV Bound on Power

Fig. 10 shows the impact of the TSV bound on WL, buffer count, and clock power consumption. These metrics are normalized to the baseline results from the single-TSV approach. The x-axis corresponds to the TSV bound used to build our multi-TSV pre-bond testable 3-D clock tree. Note that the actual TSV usage may be less than the TSV bound because the clock tree synthesis algorithm may determine that the optimal number of TSVs is less than the allowed number. For example, when the TSV bound is set to infinity, only 3097 TSVs are actually used in the four-die stack of benchmark r_5.

We first observe that the WL consistently reduces as more and more TSVs are used in our 3-D pre-bond testable clock trees. The WL savings reach 45% if the TSV bound is set to infinity. This confirms that, in general, TSVs help to reduce the overall WL of 3-D clock trees. Second, the total number of buffers (both clock buffers and TSV-buffers) increases as more TSVs are used. This is mainly due to the insertion of required TSV-buffers for pre-bond testability. Considering both trends, the power consumption decreases consistently but slowly for a time but eventually begins to rise as the cost of the TSV-buffers finally begins to outweigh the WL savings. The maximum power saving for r_5 is around 18%. The corresponding 3-D clock tree uses approximately 2500 TSVs across all four dies. With more than 2500 TSVs, the power consumption finally rises due to the excessive number of TSV-buffers. This trend gives us an optimum TSV bound for a given power budget; for the four-die stack r_5, the TSV bound should be set to 300 for a power consumption savings of 10%.

F. Impact of TSV-Buffer Insertion

As discussed earlier, TSV-buffers help trees maintain low clock skew during pre-bond test of the clock source die. Table IV shows the impact of TSV-buffer insertion, where we compare two clock trees in the clock source die, one with TSV-buffers and one without them. We observe that the skew in the source die increases 3× to 10× if TSV-buffers are not used. However, as discussed in Section V-A, TSV-buffers cause minor increases in WL and overall power consumption in the range of 2.3–11.28%.

G. Impact of Clock Source Location

Next, we consider the placement of the clock source. Table V compares two cases for a four-die stack: locating the clock source in the top die (die-0) versus in a middle die (die-1). We observe that by locating the clock source in the middle die, we use fewer TSVs while achieving comparable power consumption. The middle-die cases use 7.8–20.2% fewer TSVs than the top-die cases. Meanwhile, power and WL differences are within ± 2% in most cases.

Table VI presents a detailed list of TSV usage (stacked versus non-stacked). When connecting two clock sinks in non-adjacent dies (e.g., die-1 and die-3), we can use either two 1-stack TSVs (TSVs are not stacked) or one 2-stack TSVs (TSVs are stacked). For both cases, the number of TSVs (#TSVs) is

TABLE VII
IMPACT OF THE *cmax* (UPPER BOUND FOR THE BUFFER OUTPUT LOAD)

	Single TSV					Multi-TSV						Reduction (%)	
cmax	#Bufs	WL	Power	Skew	Slew	#TSVs	#Bufs	WL	Power	Skew	Slew	WL	Power
150	676	272 732	180.8	22.6	37.1	262	545	157 908	134.3	5.6	37.4	42.10	25.72
175	578	271 403	168.9	22.0	43.9	259	486	159 395	128.0	6.3	44.0	41.27	24.22
200	488	272 995	159.7	8.8	51.5	251	428	158 020	121.0	6.7	50.5	42.12	24.23
225	431	269 901	152.1	11.3	58.7	251	386	158 926	117.3	7.3	54.0	41.12	22.88
250	387	268 709	146.9	9.7	67.4	250	359	158 860	114.1	8.3	59.7	40.88	22.33
275	357	275 939	146.6	12.4	76.4	248	334	161 954	112.7	11.4	71.0	41.31	23.12
300	318	272 355	141.8	10.5	86.6	248	325	160 394	111.4	13.3	80.8	41.11	21.44

We use the four-die stack of r_1.

counted as two. If a clock network uses k N-stacked TSVs, the resulting #TSVs is calculated as $k \times N$. We observe that if the clock source is located in the middle, we use more 1-stack TSVs. In addition, we do not need to use 3-stack TSVs.

H. Impact of cmax on Power and Slew

Table VII shows the impact of *cmax* on WL (μm), power (mW), skew (ps), buffer count, and maximum slew (ps) as *cmax* increases from 150 fF to 300 fF. We use the four-die stack of benchmark r_1 and compare the single-TSV clock trees with the multi-TSV clock trees. We also report the reductions in WL and clock power.

We first observe that in the pre-bond testable clock tree design, bounding the maximum load capacitance for each buffer remains an efficient way to control the maximum slew. As *cmax* increases, the maximum slew in both the single-TSV and the multi-TSV cases increases. In other words, a tighter (smaller) *cmax* bound means better (smaller) slew. All of the slew values are below the 10% constraint (100 ps). Second, the power and WL benefits of the multi-TSV design remain consistent regardless of the value of *cmax*. The multi-TSV approach achieves more than 40% WL reduction and more than 21% power reduction across the full range of *cmax*. Third, for all *cmax*, a multi-TSV tree uses fewer buffers but still achieves a slightly better maximum slew. Last, clock skew is less than 30 ps for both the single-TSV and the multi-TSV cases for all values of *cmax*. There is no obvious skew trend for the single-TSV case, but the skew tends to reduce with tighter *cmax* values for the multi-TSV cases. The main reason is that the WLs are shorter in the multi-TSV cases, which results in the clock buffers added for slew control having a positive impact on delay and skew as well.

Fig. 11 shows the detailed slew distribution of the single-TSV and the multi-TSV clock trees based on the four-die stack of r_1. *cmax* is set to 300 fF. In the single-TSV case, slew varies from 12.3 ps to 86.6 ps with an average slew of 54.8 ps. The slew distribution of the multi-TSV case is 11.1–80.8 ps with an average slew value of 40.6 ps. Compared with the single-TSV case, the multi-TSV tree reduces the maximum slew and average slew by 5.8 ps and 14.2 ps, respectively, and shows a narrower slew distribution.

Fig. 12 shows the impact of *cmax* on the clock power consumption and the slew distribution (minimum, average, and maximum). We use the four-die stack implementation of r_1 for this experiment. We observe that multi-TSV designs have a positive impact on the maximum and average slew, showing nice reductions in these metrics.

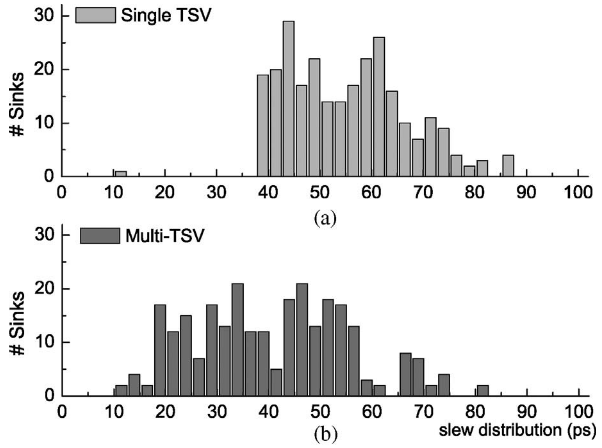


Fig. 11. Slew distribution for the four-die stack of r_1. The slew constraint is set to 10% of the clock period. *cmax* is 300 fF. (a) Single-TSV clock tree. (b) Multi-TSV clock tree with 248 TSVs. We observe that the slew values are smaller for (b).

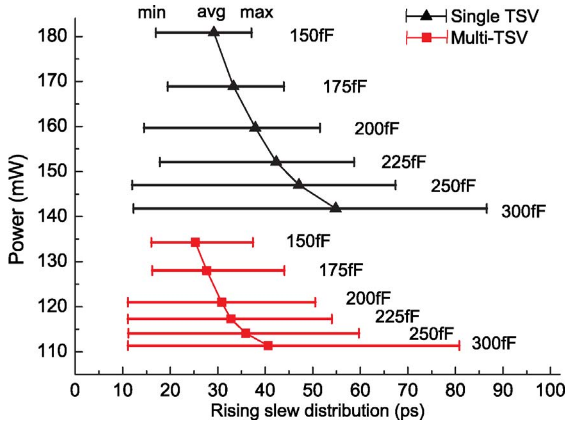


Fig. 12. Comparisons of slew variations and clock power between the single-TSV and multi-TSV clock trees based on the four-die stack of r_1. *cmax* varies from 150 fF to 300 fF.

I. Impact of TSV Capacitance

As the TSV liner oxide thickness decreases, the TSV capacitance can increase to as much as 100 fF. Table VIII shows a comparison of WL (μm), buffer count, clock power consumption (mW), and clock skew (ps) as the TSV capacitance increases from 0 fF to 100 fF. We focus on the four-die

TABLE VIII
IMPACT OF THE TSV CAPACITANCE, VARYING FROM 0 fF TO 100 fF

	Single-TSV				Multi-TSV (#TSVs = 183)						Multi-TSV (#TSVs = 2469)					
									Reduction (%)						Reduction (%)	
TSVCap	#Bufs	WL	Power	Skew	#Bufs	WL	Power	Skew	WL	Power	#Bufs	WL	Power	Skew	WL	Power
0	2939	2 312 770	1273.3	22.3	2788	2 012 360	1154.9	20.5	13.0	9.3	2970	1 337 980	972.4	23.3	42.1	23.6
15	2939	2 312 930	1272.0	22.2	2803	2 014 790	1159.1	20.3	12.9	8.9	3134	1 368 370	1041.0	20.3	40.8	18.2
25	2939	2 313 010	1272.4	21.8	2814	2 021 910	1167.4	20.7	12.6	8.3	3237	1 404 560	1087.3	18.6	39.3	14.5
50	2939	2 313 230	1273.2	21.8	2834	2 033 640	1180.8	19.9	12.1	7.3	3603	1 489 930	1220.9	21.0	35.6	4.1
100	2941	2 313 700	1274.7	19.4	2890	2 071 800	1215.0	16.0	10.5	4.7	4249	1 719 590	1499.7	25.7	25.7	−17.7

The results are normalized to the single-TSV case.

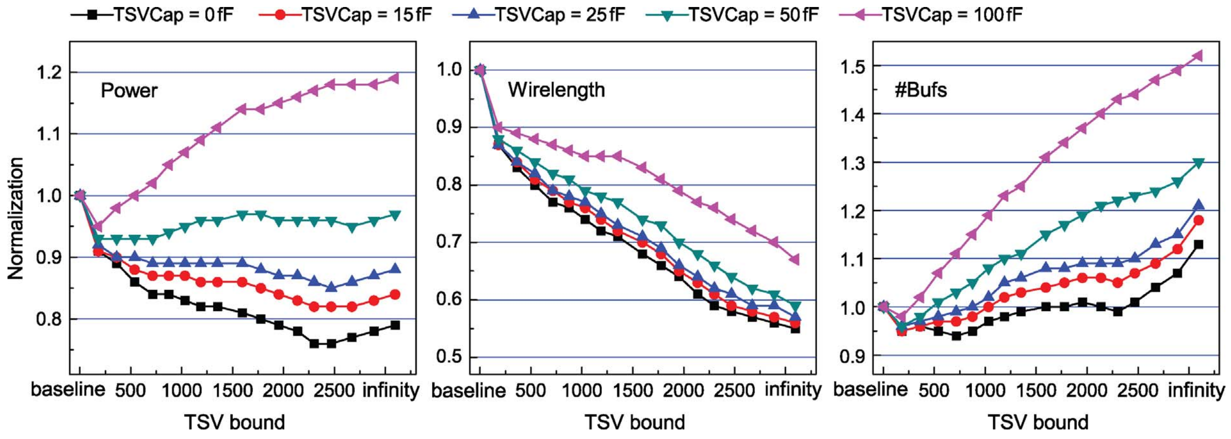


Fig. 13. Impact of the TSV capacitance and the TSV usage on the clock power consumption, WL, and buffer count trends based on the four-die stack of r_5. The baselines are the single-TSV clock tree for each value of the TSV capacitance.

stack implementation of r_5. We observe that the clock tree with 2469 TSVs has the lowest power when the TSV capacitance is small (0 fF, 15 fF, or 25 fF). The clock tree with 183 TSVs obtains the lowest power if the TSV capacitance is high (50 fF or 100 fF). Therefore, we compare three TSV-usage cases: single TSV, multi-TSV with 183 TSVs, and multi-TSV with 2469 TSVs. We also report the WL and power consumption.

We first observe that, for a fixed number of TSVs, a larger TSV capacitance leads to longer WL, more clock buffers, and higher clock power consumption. For example, for the 183 TSVs case, as the TSV capacitance increases from 0 fF to 100 fF, WL, buffer count, and clock power increase by 3.0%, 3.7%, and 5.3%, respectively.

There are two reasons for these trends. First, a larger TSV capacitance increases the difference between the internal delays of subtrees on different dies. As a result, longer wires and additional clock buffers are required to rebalance these subtrees. The larger the TSV capacitance, the longer the WL, and the greater the clock buffer count required to equalize the delays. Second, in order to meet the slew constraint, the load capacitance of each clock buffer is constrained below *cmax*. This means that as the capacitance of the clock network increases, more clock buffers must be inserted to control the slew. Therefore, more clock buffers are also required for slew control.

Our second observation from Table VIII is that the TSV capacitance diminishes the advantages of our multiple-TSV approach in terms of WL and power reduction. As the TSV capacitance increases from 0 fF to 100 fF, the WL reduction decreases from 13.0% to 10.5% for the 183 TSVs designs and from 42.1% down to 25.7% for the 2469 TSVs designs. Similarly, power savings decrease from 9.3% to 4.7% for 183 TSVs and from 23.6% to −17.7% in 2469 TSVs cases.

Finally, Table VIII shows that the TSV count and the TSV parasitics have little effect on the effectiveness of our algorithm. The clock skew is well controlled under 30 ps for all cases.

J. Trend Study: Impact of TSV Bound and Capacitance

Fig. 13 shows the impact of the TSV capacitance (TSVCap) and TSV bound on clock power, WL, and buffer count (#Bufs) trends. We use the four-die stack implementation of r_5. These metrics are normalized to the results from a design with a single column of TSVs. The TSV capacitance increases from 0 fF to 100 fF. Given both a TSVCap and a TSV bound, we construct a pre-bond testable 3-D clock tree, run SPICE simulation on the tree, and report the clock power, WL, and buffer count.

We observe that using multiple TSVs affects the clock power in different ways, which depends on the TSV capacitance. First, when the TSV capacitance is small (from 0 fF to 25 fF), we observe that using many TSVs helps to reduce the WL, buffer count, and clock power. We obtain the lowest power using 2469 TSVs. In the ideal case when using 0 fF TSVs, we can achieve up to a 23.6% power reduction compared with the single-TSV case, and WL is reduced by more than 42%. For the 15 fF or 25 fF TSVs, power is reduced by 18.2% and 14.5%, respectively.

Second, when the TSV capacitance is large (such as 50 fF or 100 fF), clock power first decreases and then increases when using more TSVs. In Fig. 13, when TSVCap is 100 fF, the lowest clock power (a 4.7% power reduction) comes from the

TABLE IX
COMPARISONS WITH [7]

	MMM-3-D+ZCTE-3-D [7]			Ours		
ckt	#TSVs	WL	Delay	#TSVs	WL	Delay
r1	83	1 441 849	1.64	74	1 567 927	1.7
r2	197	2 831 346	4.34	176	3 133 533	4.44
r3	276	3 725 294	6.37	245	4 036 177	6.89
r4	653	7 424 886	19.28	566	8 162 013	19.95
r5	1052	10 940 984	35.2	943	11 806 895	36.21

clock tree with 183 TSVs. When thousands of TSVs are used, power increases significantly.

Third, as the TSV capacitance increases, it becomes more challenging to achieve a low-power clock network. Based on 0 fF TSVs, the multi-TSV policy is able to obtain a low-power design with 23.6% power saving; for 100 fF TSVs, the multi-TSV strategy can only achieve 4.7% power reduction.

Those observations result mainly from the following factors. First, TSV usage and the TSV capacitance have opposite effects on WL: using more TSVs tends to reduce the size of each subtree in the non-clock source dies, reducing the WL. However, TSVs with large capacitance tend to unbalance the subtrees, increasing wire snaking. Depending on which factor dominates—the WL increase from the large TSV capacitance or the WL reduction from multiple TSVs—the trend of the total WL changes dramatically. The same discussion applies to the buffer count.

Last, clock power is consumed by the capacitance of the wires, buffers, and TSVs. The multi-TSV strategy helps to reduce the power consumed by the wires but at the cost of increasing the power consumed in the TSVs. When using large capacitance TSVs, the TSV power consumption increases faster than wire power consumption decreases, so the total clock power increases. Therefore, as the TSV capacitance grows, the lowest-power design is achieved with just a few TSVs. In general, a large TSV capacitance makes it hard to achieve a low-power pre-bond testable 3-D clock tree.

K. Comparison with Existing Work

In Table IX, we show the comparison of our work with [7]. Note that [7] does not support pre-bond testability, insert buffers, or provide any SPICE simulation results. However, we attempted a comparison with [7] by disabling our support for pre-bond testing and buffer insertion. We use the same benchmark settings and report the skew/delay values in the Elmore delay model. We observe that our method uses 10.4–13.3% fewer TSVs than [7] while using 7.9–10.7% more WL. Note that in our paper we can control the TSV count versus WL tradeoff by tweaking the TSV bound. In addition, these results come from unbuffered clock trees. Our pre-bond testable algorithm supports buffer insertion, which helps to properly control wire snaking and therefore better minimizes the WL.

VII. Concluding Remarks

In this paper, we demonstrated how to construct a clock tree for a 3-D-stacked IC so that both enables test of each die before bonding and provides a minimum-power clock network after bonding. Our solution utilizes many TSVs to reduce WL and clock power but necessitates the use of new circuit elements—TSV-buffers and TGs—in the clock tree to support the low-skew and low-power characteristics. We studied the impact of buffer insertion on slew rate in 3-D-stacked ICs clocking. In addition, SPICE results showed that our method of inserting multiple TSVs into the clock tree significantly reduced the WL and power consumption of the 3-D clock tree as compared against a single-TSV baseline. We also studied the impact of the TSV parasitic capacitance on power consumption and WL. It showed that a larger TSV capacitance makes it harder to optimize 3-D pre-bond testable clock trees.

Some designs allow (or even necessitate) multiple clock probe pads for each die for pre-bond test. When this happens, the test equipment must provide multiple clock probes with good OTA. If multiple clock domains are used, we will need a separate clock probe pad for each domain in each die for pre-bond testing. In this case, the WL of the redundant trees is likely to reduce because they need to connect fewer subtrees. This will likely lead to more power savings.

The authors in [9] discussed the importance of testing on the partial stacks, where testing is done not only to individual die before bonding or to the entire 3-D stack but also to the partially bonded dies. For stacks containing more than two dies, applying test after each bonding step helps to enhance the yield, but significantly increases the cost of test. In addition, the clock network in these partial stacks will suffer from high clock skew during testing. Therefore, clock delivery becomes challenging for this test method.

References

[1] X. Zhao, D. L. Lewis, H.-H. Lee, and S. K. Lim, "Pre-bond testable low-power clock tree design for 3-D stacked ICs," in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, Nov. 2009, pp. 184–190.

[2] J. Minz, X. Zhao, and S. K. Lim, "Buffered clock tree synthesis for 3-D ICs under thermal variations," in *Proc. Asia South Pacific Des. Automat. Conf.*, 2008, pp. 504–509.

[3] X. Zhao and S. K. Lim, "Power and slew-aware clock network design for through-silicon-via (TSV) based 3-D ICs," in *Proc. Asia South Pacific Des. Automat. Conf.*, 2010, pp. 175–180.

[4] *Verigy V93000 SoC Series Pin Scale Digital Cards* [Online]. Available: http://www1.verigy.com

[5] V. F. Pavlidis, I. Savidis, and E. G. Friedman, "Clock distribution networks for 3-D integrated circuits," in *Proc. IEEE Custom Integr. Circuits Conf.*, Sep. 2008, pp. 651–654.

[6] V. Arunachalam and W. Burleson, "Low-power clock distribution in a multilayer core 3-D microprocessor," in *Proc. Great Lakes Symp. VLSI*, 2008, pp. 429–434.

[7] T.-Y. Kim and T. Kim, "Clock tree embedding for 3-D ICs," in *Proc. Asia South Pacific Des. Automat. Conf.*, 2010, pp. 486–491.

[8] H.-H. S. Lee and K. Chakrabarty, "Test challenges for 3-D integrated circuits," *IEEE Des. Test Comput.*, vol. 26, no. 5, pp. 26–35, Sep.–Oct. 2009.

[9] E. J. Marinissen and Y. Zorian, "Testing 3-D chips containing through-silicon vias," in *Proc. IEEE Int. Test Conf.*, Nov. 2009, pp. 1–11.

[10] X. Wu, P. Falkenstern, and Y. Xie, "Scan chain design for 3-D integrated circuits (3-D ICs)," in *Proc. IEEE Int. Conf. Comput. Des.*, Oct. 2007, pp. 208–214.

[11] X. Wu, Y. Chen, K. Chakrabarty, and Y. Xie, "Test-access mechanism optimization for core-based 3-D SoCs," in *Proc. IEEE Int. Conf. Comput. Des.*, Oct. 2008, pp. 212–218.

[12] B. Noia, K. Chakrabarty, and Y. Xie, "Test-wrapper optimization for embedded cores in TSV-based 3-D SoCs," in *Proc. IEEE Int. Conf. Comput. Des.*, Oct. 2009, pp. 70–77.

[13] D. L. Lewis and H.-H. S. Lee, “A scan-island based design enabling pre-bond testability in die-stacked microprocessors,” in *Proc. IEEE Int. Test Conf.*, Oct. 2007, pp. 1–8.

[14] D. L. Lewis and H.-H. S. Lee, “Testing circuit-partitioned 3-D IC designs,” in *Proc. Int. Symp. VLSI*, May 2009, pp. 139–144.

[15] L. Jiang, L. Huang, and Q. Xu, “Test architecture design and optimization for 3-D SoCs,” in *Proc. Des., Automat. Test Eur.*, 2009, pp. 220–225.

[16] L. Jiang, Q. Xu, K. Chakrabarty, and T. M. Mak, “Layout-driven test-architecture design and optimization for 3-D SoCs under pre-bond test-pin-count constraint,” in *Proc. IEEE Int. Conf. Comput.-Aided Des.*, Nov. 2009, pp. 191–196.

[17] W. C. Elmore, “The transient analysis of damped linear networks with particular regard to wideband amplifiers,” *J. Appl. Phys.*, vol. 19, no. 1, pp. 55–63, 1948.

[18] K. D. Boese and A. B. Kahng, “Zero-skew clock routing trees with minimum wirelength,” in *Proc. 5th Annu. IEEE Int. ASIC Conf. Exhibit*, Sep. 1992, pp. 17–21.

[19] *RMST-Pack* [Online]. Available: http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/RSMT/RMST

[20] G. E. Tellez and M. Sarrafzadeh, “Minimal buffer insertion in clock trees with skew and slew rate constraints,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 16, no. 4, pp. 333–342, Apr. 1997.

[21] C. Albrecht, A. B. Kahng, B. Liu, I. I. Mandoiu, and A. Z. Zelikovsky, “On the skew-bounded minimum-buffer routing tree problem,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 7, pp. 937–945, Jul. 2003.

[22] C. J. Alpert, A. B. Kahng, B. Liu, I. I. Mandoiu, and A. Z. Zelikovsky, “Minimum buffered routing with bounded capacitive load for slew rate and reliability control,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 22, no. 3, pp. 241–253, Mar. 2003.

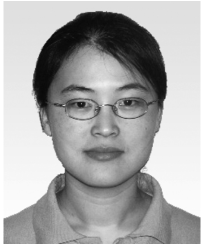
[23] S. Hu, C. J. Alpert, J. Hu, S. K. Karandikar, Z. Li, W. Shi, and C. N. Sze, “Fast algorithms for slew-constrained minimum cost buffering,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 26, no. 11, pp. 2009–2022, Nov. 2007.

[24] *GSRC Benchmark* [Online]. Available: http://vlsicad.ucsd.edu/GSRC/bookshelf/Slots/BST

[25] *ISPD Contest 2009* [Online]. Available: http://www.sigda.org/ispd/contests/ispd09cts.html

[26] *Predictive Technology Model* [Online]. Available: http://ptm.asu.edu

[27] *Synopsys Raphael* [Online]. Available: http://www.synopsys.com



Xin Zhao (S’07) received the B.S. degree from the Department of Electronic Engineering, in 2003, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2006. She is currently a Ph.D. student with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta.

Her current research interests include computer-aided design for very large scale integrated circuits, especially on physical design for low power, robustness, and 3-D integrated circuits.

Ms. Zhao was the recipient of the Best Paper Award Nomination at the International Conference on Computer-Aided Design in 2009.

Dean L. Lewis (S’02) received the B.S. degree in computer engineering and physics in 2005 from Virginia Commonwealth University, Richmond, and the M.S.E.C.E. degree from the Georgia Institute of Technology, Atlanta, in 2007. He is currently pursuing the Ph.D. degree in computer engineering under Prof. H.-H. S. Lee at the Georgia Institute of Technology.

His current research interests include design and testing for 3-D integration and resistive random-access memory. He is a member of Tau Beta Pi.

Hsien-Hsin S. Lee (M’96–SM’07) received the Ph.D. degree in computer science and engineering from the University of Michigan, Ann Arbor.

He is an Associate Professor with the School of Electrical and Computer Engineering at Georgia Institute of Technology, Atlanta. Prior to joining academia, he was a Processor Architect with Intel Corporation and later the Architecture Manager with StarCore DSP Technology Center at Agere Systems, Allentown, PA, and Motorola, Schaumburg, IL. He holds four U.S. patents. His current research interests include computer architecture, 3-D integrated circuits, low-power very large scale integration, and cyber security.

Dr. Lee was awarded the Horace H. Rackham School Distinguished Dissertation Award for his doctoral thesis at the University of Michigan. He received the DoE Early Career PI Award, the Georgia Tech ECE Outstanding Jr. Faculty Member Award, and the NSF Career Award. He has co-authored three papers that won Best Paper Awards at MICRO-33, CASES-04, and IBM PAC^2 and is serving on the Editorial Boards of *ACM Transactions on Architecture and Code Optimization* and IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He is a member of Tau Beta Pi and a senior member of the ACM.

Sung Kyu Lim (S’94–M’00–SM’05) received the B.S., M.S., and Ph.D. degrees from the Department of Computer Science, University of California, Los Angeles, in 1994, 1997, and 2000, respectively.

In 2001, he was with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, where he is currently an Associate Professor. His current research focus is on the architecture, circuit, and physical design for 3-D integrated circuits and 3-D system-in-packages.

Dr. Lim received the Design Automation Conference Graduate Scholarship in 2003 and the National Science Foundation Faculty Early Career Development Award in 2006. He was on the Advisory Board of the ACM Special Interest Group on Design Automation from 2003 to 2008 and received the ACM SIGDA Distinguished Service Award in 2008. He was an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS from 2007 to 2009 and served as a Guest Editor for the *ACM Transactions on Design Automation of Electronic Systems*. He has served the Technical Program Committee of several ACM and IEEE conferences on electronic design automation. He is a member of the Design International Technology Working Group for the 2009 renewal of the International Technology Roadmap for Semiconductors. He is the author of *Practical Problems in VLSI Physical Design Automation* (New York: Springer, 2008).