Study of Through-Silicon-Via Impact on the 3-D Stacked IC Layout

Dae Hyun Kim, *Student Member, IEEE*, Krit Athikulwongse, *Student Member, IEEE*, and Sung Kyu Lim, *Senior Member, IEEE*

***Abstract*— The technology of through-silicon vias (TSVs) enables fine-grained integration of multiple dies into a single 3-D stack. TSVs occupy significant silicon area due to their sheer size, which has a great effect on the quality of 3-D integrated chips (ICs). Whereas well-managed TSVs alleviate routing congestion and reduce wirelength, excessive or ill-managed TSVs increase the die area and wirelength. In this paper, we investigate the impact of the TSV on the quality of 3-D IC layouts. Two design schemes, namely TSV co-placement (irregular TSV placement) and TSV site (regular TSV placement), and accompanying algorithms to find and optimize locations of gates and TSVs are proposed for the design of 3-D ICs. Two TSV assignment algorithms are also proposed to enable the regular TSV placement. Simulation results show that the wirelength of 3-D ICs is shorter than that of 2-D ICs by up to 25%.**

***Index Terms*—3-D integrated chip (IC), interconnect, placement, routing, through-silicon via (TSV).**

I. Introduction

3-D INTEGRATED CIRCUITS (3-D ICs) are emerging as a promising way to overcome interconnect scaling problems of 2-D ICs and improve performance further. In 3-D ICs, gates are placed in multiple dies, and the dies are stacked vertically on top of each other as illustrated in Fig. 1. Since gates are distributed in multiple dies, the footprint area of each die of a 3-D IC becomes smaller than that of the circuit designed in 2-D. A smaller footprint area results in shorter total wirelength in 3-D ICs than in 2-D ICs [1], [2]. Therefore, 3-D ICs have a high potential to improve the performance [3], [4]. Shorter wirelength can also reduce interconnect power and improve routing congestion. Less routing congestion can in turn reduce the number of metal layers used for routing in each die of a 3-D IC, and the reduction of the metal layer count can contribute to cost reduction [5].

Vertical interconnects across dies in 3-D ICs are enabled by via-first or via-last through-silicon vias (TSVs) as shown in Fig. 1. Typical diameters of via-first TSVs range from 1 to 5 μm and those of via-last TSVs range from 5 to 20 μm [6]–[10]. To connect TSVs and other gates, metal landing pads are attached to TSVs.

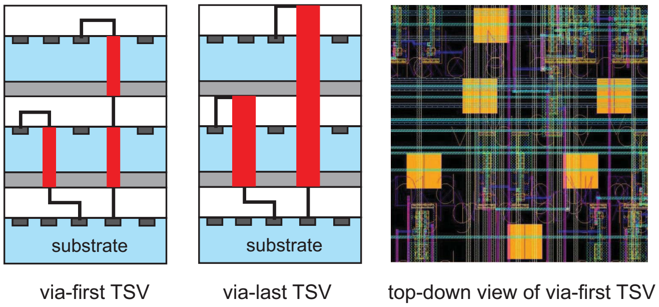


Fig. 1. Via-first and via-last TSVs with face-to-back bonding.

Although TSVs play the most important role in gate-to-gate connections across dies, TSVs have negative impact on 3-D IC designs. Above all, TSVs are fabricated in bulk silicon, so they consume silicon area, which otherwise can be used for gates. In addition, keep-out-zone rules that forbid gate placement near TSVs must be satisfied. Because of these constraints and requirements, inserting excessive amount of TSVs into 3-D ICs can cause serious area overhead. In addition, TSVs consume routing resources because TSVs need to be routed to gates or other TSVs through metal layers. This might cause routing congestion. Therefore, CAD tools for the design of 3-D ICs should carefully account for the impact of TSVs during placement and routing. However, most previous works on CAD algorithms and tools for 3-D ICs, such as [11] and [12] ignore either the sheer size of TSVs or the fact that TSVs interfere with gates and/or wires.

In this paper, the impact of TSVs on 3-D ICs is investigated. This paper is based on design rules check (DRC)-clean GDSII layouts, and a complete set of simulation results are provided. The contributions of this paper are as follows.

1) Two 3-D IC design flows, namely "TSV co-placement design flow" and "TSV site design flow," are proposed and compared. The TSV co-placement design scheme places TSVs and gates simultaneously, whereas the TSV site design scheme places TSVs at regular positions and then places gates.
2) Two TSV assignment algorithms for the TSV site scheme in which 3-D nets are assigned to pre-placed TSVs are developed. In addition, four TSV assignment algorithms are compared to investigate the impact

Manuscript received September 26, 2011; revised February 22, 2012; accepted April 22, 2012. Date of publication August 14, 2012; date of current version April 22, 2013. This work was supported in part by the National Science Foundation under Grant CCF-1018216, Grant CCF-0917000, and Grant CCF-0546382.

The authors are with the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332 USA (e-mail: daehyun@gatech.edu; krit@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/TVLSI.2012.2201760

TABLE I
ASSUMPTIONS, PARAMETERS, AND TERMINOLOGIES USED IN THIS PAPER

	Value/Meaning
Process technology	45nm
Die bonding	face-to-back
TSV shape TSV type TSV size ($n\times$)	square via-first small ($1\times$), medium ($2\times$), large ($3\times$)
$n\times$ TSV width $n\times$ TSV landing pad width $n\times$ TSV cell width	$n \cdot 1.50$ μm (n = 1, 2, or 3) $n \cdot 2.07$ μm (n = 1, 2, or 3) $n \cdot 2.47$ μm (n = 1, 2, or 3)
Min. TSV-to-cell spacing Min. TSV-to-TSV spacing	$n \cdot 0.4$ μm for $n\times$ TSV $n \cdot 0.8$ μm for $n\times$ TSV
TSV resistance ($1\times$ TSV) TSV capacitance ($1\times$ TSV)	100 mΩ 5 fF
2-D net 3-D net	a net whose cells exist in a single die a net whose cells exist across multiple dies

of the quality of TSV assignment algorithms on the wirelength.

3) Since TSVs have negative effects, such as occupying silicon area, various layouts are generated and compared to show the impact of TSVs on wirelength and area of 3-D ICs.

The rest of this paper is organized as follows. In Section II, preliminary studies on 3-D ICs and TSVs are presented. In Section III, two 3-D IC design schemes are introduced. The 3-D placement algorithm is explained in Section IV. In Section V, the TSV assignment algorithms are developed. Simulation results are shown in Section VI, and conclusions are provided in Section VII.

II. Preliminaries

In this section, previous works are discussed, and design issues, such as 3-D placement and 3-D DRC are introduced and explained. Then the impact of TSV count on die area is addressed. Table I lists assumptions, parameters, and terminologies used in this paper.

A. Previous Works

Several placement algorithms for the design of 3-D ICs have been proposed in the literature. In [13], the authors randomly place standard cells within the placement area, and use forces to move the cells in three dimensions to reduce cell overlap and temperature. In [11], the authors transform a 2-D placement result into 3-D. The proposed transformations are based on folding and stacking a 2-D design. After the transformation, they use a graph-based layer assignment method to refine the 3-D placement result by placing cells into multiple layers to reduce the number of TSVs and temperature.

Reference [12] proposes analytical and partitioning-based techniques for placement of 3-D ICs. A recursive bisection approach is used during global placement, and the cut direction for each bisection is selected as orthogonal to the largest of the width, height, or weighted depth of the placement area.

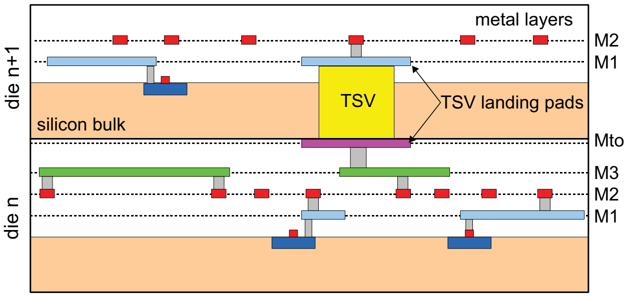


Fig. 2. TSVs, TSV landing pads, and connections to TSV landing pads.

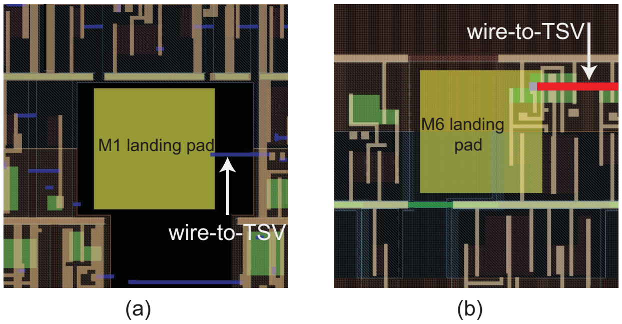


Fig. 3. TSV landing pads (yellow) and metal wires (M1 in blue and M6 in red) connected to the landing pads (Cadence Virtuoso). (a) Landing pad in M1. (b) Landing pad in M6.

In [14], the authors propose a multilevel nonlinear programming-based placement algorithm for 3-D ICs. Their objective is the weighted sum of wirelength and the number of TSVs. The authors use a density penalty function to remove overlap. They also use a bell-shaped density projection function helps to obtain a legal placement in the z-direction.

Reference [13] does not consider the TSV at all in any stage. Although [11], [12], and [14] consider the number of TSVs, all the placement algorithms do not account for TSV area.

Since the initial work of this paper was published in [15], [16], and [17] proposed different 3-D placement and TSV insertion algorithms. [16] performs global placement by iterative partitioning and then inserts TSVs. On the other hand, [17] performs global placement by analytical placement, which determines the number of TSVs. The wirelength of [17] is better than that of [15]. However, the 3-D global placer used in this paper is an improved version of the 3-D global placer used in [15], and the wirelength comparison presented in the appendix shows that the 3-D global placer of this paper is better than those of [15] and [17].

B. Design of 3-D ICs

In 3-D ICs, gates and TSVs are placed in multiple dies. Since both TSVs and gates occupy silicon area, overlaps between them should be avoided. In addition, TSVs should be routed without violating design rules. Fig. 2 illustrates connections to TSVs. Since a TSV is in fact connected to its M1 landing pad in the same die and Mtop landing pad in the bottom die (or backside landing pad in the same die), wires to these landing pads for connections to TSVs are routed. Fig. 3 shows landing pads in M1 and M6 (=Mtop) and the wires

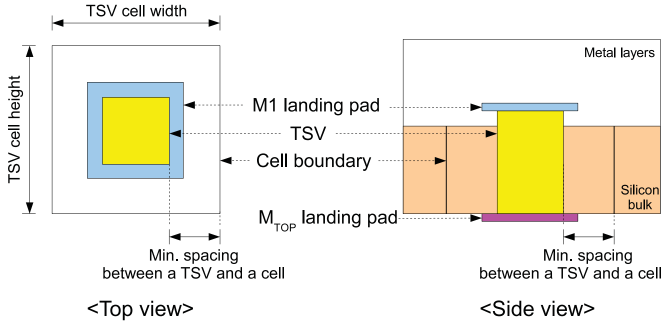


Fig. 4. Definition of a TSV cell.

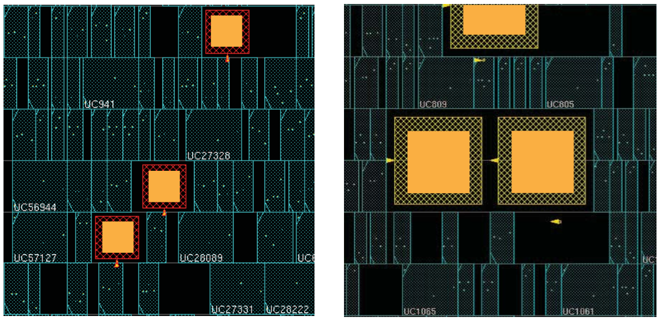


Fig. 5. 1× TSV cells (=occupying a single standard cell row) versus 2× TSV cells (=occupying two rows) in Cadence Encounter. Orange squares inside landing pads are TSVs.

connected to them in a top-down view. In case of the landing pad in Mtop in Fig. 3(b), because it is located in Mtop, it does not interfere with gates in the same die.

3-D IC layouts should also pass 3-D DRC and 3-D LVS as well as 2-D DRC and 2-D LVS. New 3-D design rules include the minimum TSV-to-TSV spacing, minimum TSV-to-cell spacing, minimum (or maximum) TSV density, and so on. In this paper, the minimum spacing rules shown in Table I are applied. 3-D LVS is checked by existing LVS tools because LVS checks logical connections.

In our design flow, TSVs are treated as cells to automate placement and routing of TSVs while optimizing locations of cells and TSVs. In order to satisfy the minimum spacing requirement around TSVs during placement, a standard cell containing a TSV landing pad in M_1 layer and whitespace around it is defined and used. This standard cell will be called a *TSV cell* for the rest of this paper. A TSV cell, shown in Figs. 4 and 5, shows 1× and 2× TSV cells placed in 3-D IC layouts. In the 1× TSV case, a TSV cell occupies 2.47 μm × 2.47 μm space and contains a landing pad and a TSV.

C. Maximum Allowable TSV Count

TSVs occupy silicon area because they are fabricated through the bulk silicon. For example, a 1× TSV cell in Table I occupies 6.1 μm^2. However, a 1× two-input NAND gate of the NCSU 45-nm library [20] occupies 1.88 μm^2, and a 1× D flip-flop gate of the Nangate 45-nm library [21] occupies 4.52 μm^2. Therefore, ignoring TSV area leads to serious underestimation of the additional area for TSV insertion.

Since the smallest 2-D chip area is simply the total cell area, the maximum TSV count such that the chip area of a 3-D IC becomes smaller than a pre-determined number can be computed. The maximum TSV count, $N_{\text{TSV}_{\max}}$, based on 2-D and 3-D chip areas can be calculated by the following equations:

$$N_{\text{TSV}_{\max}} = \frac{(A_{\text{3-D}} - A_{\text{2-D}})}{A_{\text{TSV}}} \tag{1}$$

where $A_{\text{3-D}}$ is the sum of the area of all dies of a 3-D IC, $A_{\text{2-D}}$ is the die area when the circuit is designed in 2-D, and A_{TSV} is the area required by a TSV. $N_{\text{TSV}_{\max}}$ is the maximum number of TSVs that can be used in 3-D ICs.

D. Minimum TSV Count

While the maximum allowable TSV count constrains the maximum number of TSVs that can be used in 3-D ICs, the minimum TSV count provides the feasibility of the 3-D IC design in terms of the TSV count because the minimum TSV count could be greater than the maximum allowable TSV count in some cases.

The minimum TSV count can be estimated by running min-cut partitioning because a 3-D net spanning in K dies needs at least $K - 1$ TSVs (one TSV between two adjacent dies). Table II lists the minimum number of TSVs estimated by *k-way* min-cut partitioning using hMetis [18] and area overhead. As the table shows, some circuits (e.g., AL1) have huge area overhead (up to 34.15%) caused by TSVs,[1] so they could not be designed in four dies if the maximum allowable area overhead is small (e.g., 10%). However, this area overhead is strongly dependent on the TSV size and the average cell area. If a circuit consists of many large cells or an older process, such as 0.18 μm is used, the average cell area of the circuit will be much larger than the total TSV area. In this case, the area overhead caused by TSVs becomes small.

E. Irregular Versus Regular TSV Placement

Placing TSVs irregularly and placing TSVs regularly have their own pros and cons. Irregular TSV placement is expected to produce layouts having shorter wirelength than layouts generated by regular TSV placement because irregular TSV placement has higher degree of freedom than regular TSV placement with respect to TSV locations. However, regular TSV placement has better exposure quality of the lithographic process [22], smaller TSV stress [23], and better metal layer density [24]. Moreover, regular placement of die-to-die connections are already being used in some 3-D process technologies [25] for better die-to-die bonding.

III. 3-D IC Design Flow

Two 3-D IC design flows are devised in this paper, namely TSV co-placement and TSV site, as illustrated in Fig. 6. These flows are developed in such a way that existing 2-D routing tools can be used while TSVs are handled effectively. Utilizing existing 2-D routing tools makes it easy to generate GDSII

[1] The area overhead caused by a TSV is due to the TSV, liner around the TSV, and keep-out zone. In this paper, A_{TSV} of a 1×TSV is 6.1009 μm^2.

TABLE II

BENCHMARK CIRCUITS AND THEIR PARTITIONING RESULTS. hMETIS [18] IS USED FOR k-WAY MIN-CUT PARTITIONING. CELL AREA DENOTES THE SUM OF THE AREA OF ALL THE CELLS, AVG. CELL AREA DENOTES THE AVERAGE CELL AREA, AND TSV AREA DENOTES THE SUM OF THE AREA OF ALL TSVS. THE AREA OF 1× TSVS ARE USED. IN THE PROFILE, "AL" DENOTES AN ARITHMETIC LOGIC AND "μP" DENOTES A MICROPROCESSOR. BENCHMARK CIRCUITS MARKED WITH "*" ARE FROM IWLS 2005 BENCHMARK SUITES [19], AND OTHER BENCHMARK CIRCUITS ARE FROM INDUSTRY

Circuit	No. of gates	No. of nets	Total cell area (mm^2)	Avg. cell area (μm^2)	No. of dies						Profile
					2		3		4		
					Min. no. of TSVs	$\frac{\text{TSV area}}{\text{Cell area}}$	Min. no. of TSVs	$\frac{\text{TSV area}}{\text{Cell area}}$	Min. no. of TSVs	$\frac{\text{TSV area}}{\text{Cell area}}$	
AL1	15 K	15 K	0.033	2.236	253	4.57%	759	13.70%	1, 892	34.15%	AL
AL2	30 K	30 K	0.082	2.754	964	7.07%	1, 535	11.26%	2, 910	21.34%	AL
AL3*	77 K	77 K	0.227	2.937	733	1.94%	956	2.53%	2, 019	5.34%	IP core
AL4*	109 K	109 K	0.257	2.350	1, 502	3.51%	2, 480	5.80%	3, 639	8.51%	AL
AL5	324 K	328 K	1.163	3.585	452	0.23%	334	0.17%	1, 107	0.57%	AL
AL6	445 K	484 K	1.625	3.656	2, 574	0.97%	5, 591	2.10%	6, 384	2.40%	AL
AL7	661 K	676 K	2.404	3.635	503	0.13%	818	0.21%	2353	0.60%	AL
MP1	16 K	16 K	0.039	2.460	87	1.35%	465	7.22%	542	8.41%	μP
MP2	20 K	20 K	0.048	2.433	346	4.29%	821	10.18%	942	11.68%	μP
MP3*	88 K	89 K	0.246	2.787	493	1.20%	1, 166	2.84%	1, 689	4.11%	μP
MP4*	104 K	104 K	0.264	2.545	168	0.38%	363	0.83%	707	1.61%	μP cores
MP5*	169 K	169 K	0.463	2.740	54	0.07%	409	0.53%	362	0.47%	μP cores

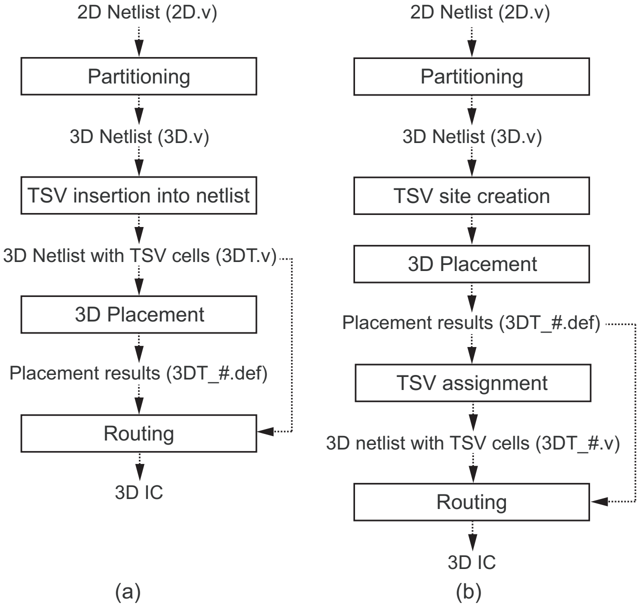


Fig. 6. Two 3-D IC design flows developed in this paper. (a) TSV co-placement. (b) TSV site.

layouts of 3-D ICs for in-depth analysis. Notice that the design flow is for via-first type TSVs.[2]

A. Partitioning

A way to perform 3-D global placement using force-directed placement algorithms is to add an axis along the z-direction. In this case, the quadratic wirelength function is expressed as

$$\Gamma = \Gamma_x + \Gamma_y + \Gamma_z \tag{2}$$

where Γ_x, Γ_y, and Γ_z are wirelengths along the x-, y-, and z-axis, respectively, and each component of Γ is minimized independently. However, this method cannot place cells in multiple dies unless the initial placement algorithm intentionally places cells in multiple dies. The reason is because all I/O pins are in the topmost die (die 0), so if cells are placed in the topmost die at the initial placement step, they will not be spread across multiple dies. Therefore, partitioning is used as a pre-process for 3-D global placement in our flow and across-die movement of cells during 3-D global placement is not allowed.

In the first step of both design schemes, a modified FM partitioning algorithm [26] is used to distribute cells in a 2-D netlist into N_{die} dies. After partitioning, die order should be determined. For instance, suppose that a 3-D IC is designed in four dies. Assuming I/O pins are placed in die 0, there exist six (=3!) different die orderings. In this paper, however, any particular die ordering scheme is not used. Instead, each partition pX is treated as die X (e.g., $p0$ becomes die 0). Die ordering affects the TSV count, but the total number of TSVs varies actually in a small range for different die orderings.

B. TSV Insertion and Placement

In the TSV co-placement scheme, TSVs are added into the 3-D netlist during the TSV insertion step, and then cells and TSVs are placed simultaneously during 3-D placement. The 3-D placement algorithm is explained in detail in Section IV.

In the TSV site scheme, on the other hand, TSVs are pre-placed uniformly on each die in the TSV site creation step, and then cells are placed in the 3-D placement step. During 3-D placement, pre-placed TSVs are treated as placement obstacles because TSVs should not be overlapped with any cell. After 3-D placement, TSV assignment is performed to determine which 3-D nets use which pre-placed TSVs. Then, the 3-D netlist is updated to insert the assigned TSVs into

[2] Handling via-last type TSVs requires additional steps. For the TSV co-placement flow, refinement of TSV locations after global placement can remove overlaps between two TSVs in adjacent dies. For the TSV-site flow, overlaps between two TSVs in adjacent dies can be avoided by using different TSV array sizes.

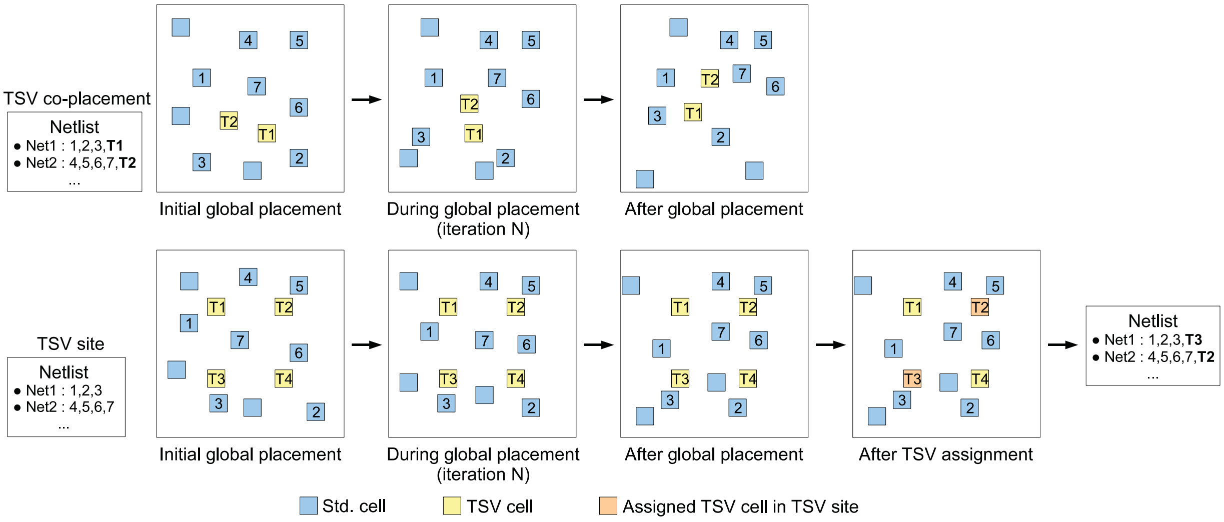


Fig. 7. TSV insertion, 3-D placement, TSV assignment, and netlist generation.

the netlist. Fig. 7 illustrates the TSV co-placement and TSV site schemes. For detailed placement, the detailed placer of Cadence system-on-chip Encounter is used [27].

C. Routing

After 3-D placement, the placement result is dumped into DEF files and a netlist file is generated for each die. At this time, TSV landing pads should be made at both ends of a TSV as shown in Fig. 2. While an M_1 landing pad is placed in die($n+1$), its corresponding M_{TOP} landing pad is placed in die(n) at the same location. An M_{TOP} landing pad in die(n) is represented by placing a pin in the DEF file of die(n) and adding the pin into the netlist of die(n). Then, Encounter is used to route each die.

IV. 3-D Global Placement Algorithm

The 3-D global placement algorithm used in this paper is based on a force-directed quadratic placement algorithm [28]. The algorithm is modified to place cells and TSVs in 3-D.

A. Overview of Force-Directed Quadratic Placement

In quadratic placement, optimal locations of cells are computed by minimizing the quadratic wirelength function Γ, expressed as

$$\Gamma = \Gamma_x + \Gamma_y \tag{3}$$

where Γ_x and Γ_y are wirelengths along the x- and y-axis. Γ_x is written as

$$\Gamma_x = \frac{1}{2}\mathbf{x}^{\mathrm{T}}\mathbf{C}_x\mathbf{x} + \mathbf{x}^{\mathrm{T}}\mathbf{d}_x + \text{constant} \tag{4}$$

where $\mathbf{x}$ represents the x-position of N cells, $\mathbf{C}_x$ is an $N \times N$ connectivity matrix using the bound-to-bound net model [28], and $\mathbf{d}_x$ represents the connectivity between cells and pins. Element $c_{x,ij}$ of $\mathbf{C}_x$ is the weight of the connection between cell i and cell j, and element $d_{x,i}$ is the negative weighted position of the fixed pins connected to cell i. Γ_x is minimized by solving the following equations:

$$\nabla_x \Gamma_x = \mathbf{C}_x\mathbf{x} + \mathbf{d}_x = \mathbf{0}. \tag{5}$$

Quadratic placement can be viewed as an elastic spring system when Γ is treated as the total spring energy of the system. Because the derivative of the spring energy is a force, the derivative of Γ_x in (4) can be viewed as a net force $\mathbf{f}_x^{\text{net}}$ as

$$\mathbf{f}_x^{\text{net}} = \nabla_x \Gamma_x = \mathbf{C}_x\mathbf{x} + \mathbf{d}_x \tag{6}$$

where ∇_x is the vector differential operator. At equilibrium, $\mathbf{f}_x^{\text{net}}$ is zero and Γ_x is minimized, which results in overlaps among cells. To remove cell overlap, [28] uses two additional forces, the move force $\mathbf{f}_x^{\text{move}}$ and the hold force $\mathbf{f}_x^{\text{hold}}$.

The move force is a density-based force that spreads cells away from high cell density area to low cell density area to reduce cell overlap. The move force in [28] is defined for 2-D ICs, thus it is modified to lower cell densities in 3-D ICs. The modification is explained in Section IV-C.

The hold force is used to decouple each placement iteration from its previous iteration. It cancels out the net force that pulls cells back to the location in the previous iteration. The hold force is written as

$$\mathbf{f}_x^{\text{hold}} = -(\mathbf{C}_x\mathbf{x}' + \mathbf{d}_x) \tag{7}$$

where $\mathbf{x}'$ contains the x-position of cells from the previous placement iteration. When no move force is applied, the hold force holds cells at their current locations.

The total force $\mathbf{f}_x$ is the summation of the net force, move force, and hold force. The total force is set to zero to minimize wirelength while removing cell overlap.

B. Overview of Our 3-D Placement Algorithm

The proposed 3-D placement algorithm is divided into three phases: initial placement, global placement, and detailed placement. In the first phase, the initial cell locations are computed by solving (5). In the second phase, the amount of cell overlaps is reduced by applying the move force and the hold force included in solving the equation. Overlaps are removed gradually because moving cells rapidly degrades the overall placement quality. Global placement continues until the amount of remaining cell overlap becomes lower than a pre-determined overlap ratio. Then detailed placement is performed using the detailed placer included in Cadence encounter.

C. Cell Placement in 3-D ICs

A major extension on the 2-D force-directed quadratic placement algorithm in this paper is to modify the move force in [28] so that removing cell overlap is performed in each die separately. For example, the move force is not applied between two cells at the same x and y location if they are in different dies.

The placement problem is formulated as a global electrostatic problem, by treating cell area as positive charge and chip area as negative charge. The placement density D on die d can be computed by

$$D(x,y)\Big|_{z=d} = D^{\text{cell}}(x,y)\Big|_{z=d} - D^{\text{chip}}(x,y)\Big|_{z=d} \tag{8}$$

where $D^{\text{cell}}(x,y)\big|_{z=d}$ is the cell density at position (x,y) in die d, and $D^{\text{chip}}(x,y)\big|_{z=d}$ is the chip capacity at position (x,y) in die d.

After D is computed, the following Poission's equation is solved to compute the placement potential Φ as follows:

$$\Delta\Phi(x,y)\Big|_{z=d} = -D(x,y)\Big|_{z=d}. \tag{9}$$

The move force is modeled by connecting cell i to its target point $\mathring{x}_i$ with a spring of spring constant $\mathring{w}_i$. The target point is computed by

$$\mathring{x}_i = x_i' - \frac{\partial}{\partial x}\Phi(x,y)\Big|_{(x_i',y_i'),z=d} \tag{10}$$

where x_i' is the x-position of cell i being placed on die d in the previous placement iteration. The spring constant is initially defined by

$$\mathring{w}_i = \frac{A_i}{A^{\text{cell}}\big|_{z=d}} \tag{11}$$

where A_i is the area of cell i, and $A^{\text{cell}}\big|_{z=d}$ is total area of cells being placed on die d. Then, the spring constant is adjusted iteratively using the quality control mechanism in [28]. Therefore, for cell i, the move force is $f_{x,i}^{\text{move}} = \mathring{w}_i(x_i - \mathring{x}_i)$, where x_i is the x-position of cell i. The move force $\mathbf{f}_x^{\text{move}}$ is finally defined for 3-D ICs by

$$\mathbf{f}_x^{\text{move}} = \mathring{\mathbf{C}}_x(\mathbf{x} - \mathring{\mathbf{x}}) \tag{12}$$

where $\mathring{\mathbf{C}}_x$ is a diagonal matrix of $\mathring{w}_i$, $\mathbf{x}$ is a vector representing the x-position of N cells being placed, and $\mathring{\mathbf{x}}$ is a vector representing the target x-position of the cells.

D. Pre-Placement of TSVs in TSV Site Scheme

In the TSV site scheme, TSVs are placed evenly and then cells are placed. Therefore, TSVs are treated as placement obstacles during cell placement. The number of TSVs in each row and column is computed by

$$N_{\text{TSV}_d} = N_{\text{TSV}_{d,\min}} \times K_{\text{TSV}} \qquad K_{\text{TSV}} \geq 1 \tag{13}$$

$$N_{\text{TSV}_{d,\text{row}}} = \lfloor\sqrt{N_{\text{TSV}_d}}\rfloor \tag{14}$$

$$N_{\text{TSV}_{d,\text{col}}} = \left\lceil \frac{N_{\text{TSV}_d}}{N_{\text{TSV}_{d,\text{row}}}} \right\rceil \tag{15}$$

where $N_{\text{TSV}_{d,\min}}$ is the minimum number of TSVs on die d, and K_{TSV} is a multiplying factor for the number of TSVs. If K_{TSV} is greater than one, more TSVs than the minimum TSV count are placed to increase the selectivity during TSV assignment.

Placement obstacles can be handled naturally by means of the placement density in [28]. By including the area of pre-placed TSVs in the computation of the placement density, the move force is altered in such a way that it moves cells being placed away from pre-placed TSVs. The area of pre-placed TSVs is also included in (8).

V. TSV Assignment

The TSV assignment problem in the TSV site scheme is to assign 3-D nets to TSVs for given sets of dies, 3-D nets, placed cells, and placed TSVs while optimizing objective functions, such as the total wirelength of the 3-D nets. Constraints in the TSV assignment problem are as follows:

1) a TSV cannot be assigned to more than one 3-D net;
2) a 3-D net should use at least one TSV.

A. Optimum Solution for TSV Assignment

The authors of [29] show the binary integer linear programming (BILP) formulation to find the optimum solution of the TSV assignment problem for two dies. Since the number of binary integer variables in the formula is too big, they also introduce and develop heuristic algorithms, an approximation method based on the Hungarian method [30], and a neighborhood search method.

If a 3-D IC is designed in more than two dies and a 3-D net spans more than two dies, all the combinations of TSVs in different dies should be taken into account for the cost computation. In Fig. 8(a), for example, the 3-D net is assigned to T_1 in die 1 and T_6 in die 2, and the cost (=wirelength) is approximately $2L$. However, in Fig. 8(b), the 3-D net is assigned to T_3 in die 1 and T_6 in die 2, and the cost is approximately L. Although T_6 is used in both cases, its contribution to the cost is different. Therefore, the cost should be computed for each combination of TSVs in different dies.

The optimum solution for a TSV assignment problem for more than two dies is found by the following formulation:

min

$$\sum_{i=1}^{N_{3\text{-DNet}}} \sum_{k=1}^{CB_i} \sum_{p=1}^{N_{\text{TSV}}} d_{i,k,p} \cdot x_{i,k,p} \tag{16}$$

s.t

$$\sum_{k=1}^{CB_i} \sum_{p=1}^{N_{\text{TSV}}} x_{i,k,p} = N_{\text{die}} - 1 \quad (i = 1, \ldots, N_{3\text{-DNet}}) \tag{17}$$

$$\sum_{i=1}^{N_{3\text{-DNet}}} \sum_{k=1}^{CB_i} x_{i,k,p} \leq 1 \quad (p = 1, \ldots, N_{\text{TSV}}) \tag{18}$$

where N_{die} is the number of dies, $N_{3\text{-DNet}}$ is the total number of 3-D nets, CB_i is the total number of combinations of TSVs for the 3-D net H_i, N_{TSV} is the total number of

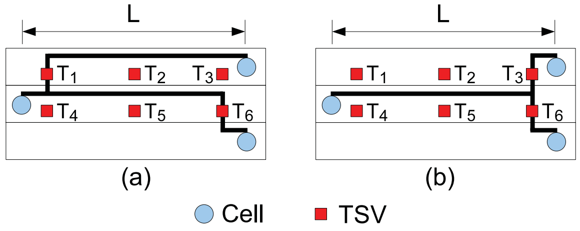


Fig. 8. Cost computation for different combinations of TSVs in three dies (side view). (a) Wirelength $= 2L$ when T_1 and T_6 are selected. (b) Wirelength $= L$ when T_3 and T_6 are selected.

TSVs, and $d_{i,k,p}$ is the cost when the pth TSV in the kth combination is used for the 3-D net H_i. Here, $x_{i,k,p}$ is 1 if: 1) the 3-D net H_i uses the kth combination and 2) the kth combination uses the TSV T_p, is 0 otherwise. Equation (17) denotes that a 3-D net uses only one combination of TSVs and (18) denotes that a TSV is assigned to at most one 3-D net.

The number of variables in this problem is also very large. Even if TSVs available for a 3-D net are limited to TSVs inside a small window, the number of combinations is still large. For example, if a 3-D net spans in four dies, and the window contains 20 TSVs in each die, 8000 combinations exist for the net. Moreover, restriction on the window size may result in infeasibility of BILP. Therefore, two heuristic algorithms are proposed in the next two sections.

B. Minimum Spanning Tree (MST)-Based TSV Assignment

In this method, the MST algorithm is used for TSV assignment as shown in Algorithm 1. First, 3-D nets are sorted in the ascending order of their bounding box size. Since a 3-D net whose bounding box is large contains more TSVs to choose from, higher priority is given to 3-D nets having smaller bounding boxes. After sorting, an MST is constructed using Kruskal's algorithm for each 3-D net, and the edges of the MST are sorted in the ascending order of their lengths because a short edge means short wirelength. After constructing the MST and sorting edges, each edge is checked in the ascending order. If the edge spans over two adjacent dies that are not connected yet, the unassigned TSV closest to the edge is chosen and marked as "assigned to this net." This process is repeated until all dies in the 3-D net are connected by TSVs. During this assignment process, the distance between an available TSV and a 3-D edge is computed as follows. The edge is projected to a 2-D plane so that the 3-D edge becomes a 2-D segment. Then, the distance is computed by calculating the Manhattan distance between the TSV and its nearest point in the segment.

An example is shown in Fig. 9. The shortest edge in the figure is the vertical edge connecting dies 1 and 2. The available TSV closest to the edge is T_3, so it is assigned to the 3-D net. Since this 3-D net spans from die 1 to die 3, a TSV in die 2 is needed to connect cells in die 2 and die 3. The TSV closest to the next shortest edge is T_6. Since this is an unavailable TSV, which means it has already been assigned to another net, the next closest TSV, T_5 is found and assigned.

Algorithm 1 MST-Based TSV Assignment

```
Sort 3-D nets in the order of bounding box size
for each 3-D net ($H_i$) do
  construct an MST (Kruskal's algorithm)
  sort edges in the ascending order of edge length
  for each edge ($e_j$) do
    if this $e_j$ connects two dies already connected then
      continue;
    end if
    find the TSV nearest to $e_j$ in the same die
    assign this TSV to $H_i$
    if all dies are connected then
      break;
    end if
  end for
end for
```

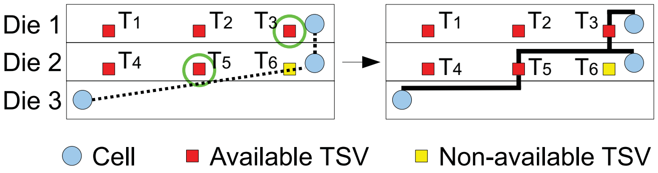


Fig. 9. Example: MST-based TSV assignment (side view).

C. Placement-Based TSV Assignment

The second TSV assignment method is based on 3-D placement. In this method, the the assignment problem is solved by a 3-D placement algorithm. Algorithm 2 shows the placement-based TSV assignment algorithm.

After placing gates in the 3-D placement stage, the placed gates are converted into pins in a new 3-D netlist. Therefore, there exist only pins, which are actually I/O pins and placed gates, in the netlist. Then, movable TSVs are inserted into this netlist, and TSV co-placement is performed. After placement is finished, TSV locations are loaded from the 3-D placement result. At this time, however, the TSVs inserted in the netlist are assigned to pre-placed TSVs. Fig. 10 shows an example. In the first step, movable TSVs (gray squares) are inserted into a new netlist and placed by the 3-D placement algorithm. After 3-D placement, the final locations of the movable TSVs are assigned to their nearest pre-placed TSVs. The final assignment result is shown in the rightmost figure of Fig. 10.

VI. Simulation Results

IWLS 2005 benchmarks [19] and several industrial circuits are used for layout generation. Table II shows the benchmark circuits and their details. The NCSU 45-nm technology [20] is used for the process technology. The 3-D placer and TSV assignment programs are implemented using C/C++ and Intel math kernel library 10.0 is used for matrix computation.

Algorithm 2 Placement-Based TSV Assignment

```
Convert all the placed gates into fixed pins
insert movable TSVs into 3-D nets in each die
run TSV co-placement
load the 3-D placement result
for each TSV ($T_m$) do
    find the nearest available TSV ($T_s$) in TSV sites
    assign $T_s$ to $T_m$
end for
```

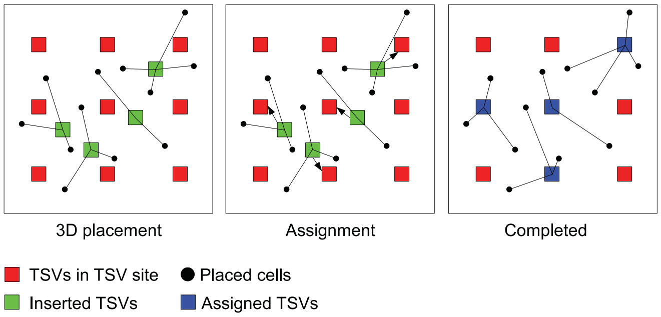


Fig. 10. Example: placement-based TSV assignment (top view).

The system has Intel Xeon 2.5 GHz CPUs with 16 GB memory and 64-bit Linux OS. Fig. 11 shows two representative layouts designed by the TSV co-placement and TSV site schemes.

A. Wirelength and Runtime Comparison

Table III shows wirelength, die area, and runtime of 2-D and 3-D placement results. For 2-D placement, the 3-D placer is run in a 2-D mode in which partitioning is not executed. For 3-D placement, the TSV co-placement scheme is used with four dies and 1×TSVs.

As to wirelength, wirelength of all the circuits except MP5 is reduced. In the MP5 case, the wirelength of the four-die 3-D implementation is almost the same as that of its 2-D implementation. Except MP5, the amount of wirelength reduction in nonmicroprocessor circuits (AL1 to AL7) is 1% to 25%, but the amount of wirelength reduction in microprocessor circuits (MP1 to MP4) is 1% to 10%.

To figure out why nonmicroprocessor circuits have more wirelength reduction than microprocessor circuits, wirelength distributions of AL4 and MP5 are shown in Fig. 12. In Fig. 12(a), long interconnections of AL4 in the 2-D design become shorter in the 3-D design. The longest wire in the 2-D design of AL4 is about 900 μm long, whereas that in the 3-D design is about 310 μm long. This effect is due to the smaller footprint area and connections in the z-direction by TSVs.

The wirelength distribution of the 2-D design of MP5 is very similar to that of the 3-D design of MP5 as shown in Fig. 12(b). The lengths of the longest wires in the 2-D and the 3-D designs are also similar. Therefore, even if multiple dies are stacked, the total wirelength does not change. Table II also supports this analysis. MP5 (0.463 mm^2) is larger than AL4 (0.257 mm^2), but the longest wire of MP5 is shorter than that of AL4 (730 μm versus 900 μm). Actually, the die width of MP5 implemented in 3-D is 410 μm, so the corner-to-corner Manhattan distance is 820 μm, which is longer than the longest wire (730 μm). However, the die width of AL4 implemented in 3-D is 310 μm, so the corner-to-corner Manhattan distance is 620 μm, which is shorter than the longest wire (900 μm). Therefore, AL4 benefits from 3-D implementation, but MP5 does not.

This is also related to the min-cut partitioning result shown in Table II. For example, the min-cut size of AL4 in 2-way partitioning is 1502 out of 109 K nets while that of MP5 is 54 out of 169 K nets. This means that MP5 is a highly modularized circuit, so it does not benefit from 3-D implementation with respect to wirelength.

Regarding runtime, 3-D placement in general needs shorter runtime than 2-D placement.[3] The reason is that an initial 3-D placement of a circuit is likely to have fewer overlaps than an initial 2-D placement of the circuit because each die in a 3-D IC has fewer cells to be placed. Since the force-directed quadratic placement algorithm spends a significant portion of its runtime in overlap removal, having fewer cells in a die improves runtime. In Table III, the 3-D global placement is 1.3× to 5× faster than the 2-D global placement.

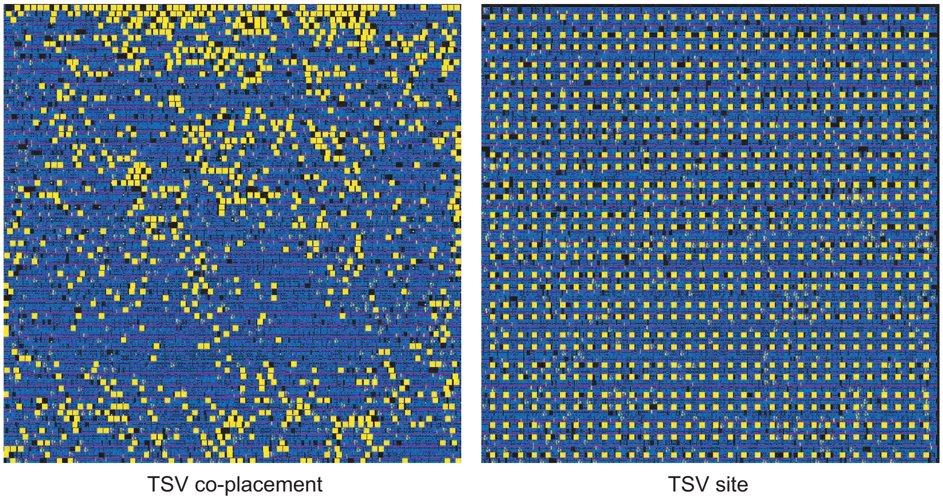
Since the design of an IC needs routing as well as placement, the runtime for routing is also compared. The 3-D placement generates N_{die} placement results. Therefore, global and detailed routing for each die is run concurrently. Then, the runtime for routing of a 3-D IC is obtained by choosing the maximum runtime. In this simulation, the ratio between the runtime for routing of 2-D ICs and the runtime for routing of 3-D ICs is between 2.73 and 5.11. The runtime for routing of 3-D ICs is much smaller than that for routing of 2-D ICs because the area of each die of a 3-D IC is smaller than that of its 2-D counterpart.

B. Metal Layers and Silicon Area Comparison

Since each die of a 3-D design has fewer cells than a 2-D design, the number of metal layers required for 3-D designs could be less than that for 2-D designs. Therefore, the minimum number of metal layers required to route all dies successfully is found and compared. For a fair comparison, the same area utilization is used for both 2-D and 3-D designs. The "# ML" columns in Table III list comparisons of the minimum number of metal layers in 2-D and 3-D designs. Except for AL5, AL6, and AL7, all circuits are routable with four metal layers in their 3-D designs, but the 2-D designs of AL2, AL4, AL5, AL6, AL7, MP4, and MP5 are not routable with four metal layers because of high routing congestion.

Table III also shows the area overhead of 3-D IC layouts. For small circuits, the area overhead is large (6% to 29%). However, the area overhead in large circuits is relatively small (2% to 16%). Since the area overhead is determined by the number of TSVs, if few TSVs are used for a small design, its area overhead could be negligible. Likewise, if too many TSVs are used for a large design, the area overhead could be significant.

[3] Partitioning used as a pre-process of 3-D placement for 3-D ICs needs a small fraction (0.05% to 2.5% in our simulation) of runtime for 3-D placement.


TSV co-placement

TSV site

Fig. 11. Cadence Virtuoso snapshot of the bottommost die of AL1 designed by TSV co-placement and TSV site schemes. Bright squares are TSVs.

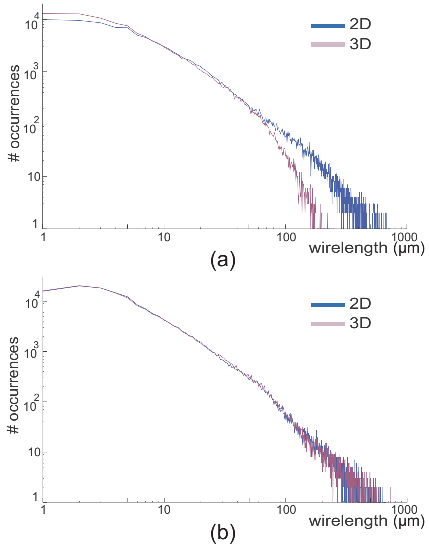


Fig. 12. Wirelength distribution of (a) nonmicroprocessor circuit (AL4) whose die width is 605 μm in a 2-D design and 310 μm in a 3-D design (4 dies) and (b) microprocessor circuit (MP5) whose die width is 812 μm in a 2-D design and 410 μm in a 3-D design (4 dies).

The area of 3-D designs is always larger than that of 2-D designs in the simulation. However, the area of a 2-D design could be larger than that of its 3-D design. As seen in Table III, some 2-D designs are not routable with four metal layers. Therefore, if there is a tight constraint on the available metal layers (e.g., four metal layers), the 2-D design not routable under the constraint should be expanded. In this case, the area of a 2-D design could be larger than that of its 3-D design.

C. On Wirelength and Die Area Versus Number of Dies

As the number of dies increases, the footprint area tends to decrease,[4] so the wirelength is expected to decrease while the total die area is expected to increase. Therefore, how wirelength and die area vary when the die count increases are observed in this section. In this simulation, the TSV co-placement scheme and 1 × TSVs are used.

Table IV lists wirelength, die area, runtime, and the number of TSVs when the die count varies from two to four. As the die count goes up, in general, the number of TSVs increases, but the wirelength decreases for the nonmicroprocessor circuits. For further experiment on this, the number of dies (N_{die}) is varied from 2 to 16, and wirelength, die area, and the number of TSVs are observed for AL4 in Fig. 13. The wirelength of AL4 dramatically decreases as N_{die} increases from two to five, then it generally goes up. If N_{die} increases further, the TSV count and the die area increase as shown in Fig. 14. In other words, increasing N_{die} is helpful at first, but becomes harmful at a larger number of dies because the TSV count increases as N_{die} goes up, which increases die area.

D. TSV Co-Placement Versus TSV Site

Table V lists the wirelength of five different placement schemes: the TSV co-placement (IR), the MST-based TSV site placement (R-MST), the placement-based TSV site placement (R-PL), the neighborhood search-based TSV site placement (R-NS, [29]), and the network flow-based TSV site placement (R-NF, [31]). The TSV co-placement (IR) designs always show shorter wirelength than the TSV site placement designs. The amount of wirelength reduction of IR as compared to R-MST, R-PL, R-NS, and R-NF is approximately 4%, 8%, and 10% on average in two-die, three-die, and four-die implementations, respectively. A reason that the TSV co-placement scheme produces shorter wirelength than the TSV site placement schemes is because the TSV co-placement scheme optimizes TSV locations and cell locations simultaneously, while pre-placed TSVs in the TSV site schemes obstruct optimal gate placement.

Table VI lists the additional runtime required for TSV assignment in R-MST and R-PL. Since TSV assignment is an additional process after 3-D placement, runtime for 3-D placement is also reported in IR and R columns. TSVs are

[4] When the number of dies increases, if the TSV area is ignored, the footprint area monotonically decreases. However, the number of TSVs has a great effect on the footprint area. If too many TSVs are used in a particular partitioning case, the footprint area at that die count could increase.

TABLE III

COMPARISON OF WIRELENGTH (WL), DIE AREA (AREA), THE MINIMUM NUMBER OF METAL LAYERS (# ML) REQUIRED TO ROUTE ALL DIES SUCCESSFULLY, AND RUNTIME (RUNTIME) FOR 2-D AND 3-D PLACEMENT. FOR 2-D PLACEMENT, THE PLACER IS RUN WITHOUT PARTITIONING. FOR 3-D PLACEMENT, FOUR-DIE IMPLEMENTATION, TSV CO-PLACEMENT SCHEME, AND 1 × TSV ARE USED

ckt	2-D				3-D				
	WL (um)	Area (μm^2)	# ML	Runtime (s)	WL (μm)	Area (μm^2)	# ML	Runtime (s)	# TSVs
AL1	257 051 (1.0)	48 400 (1.0)	4	143 (1.0)	243 610 (**0.95**)	62 500 (1.29)	4	51 (**0.36**)	1305
AL2	631 818 (1.0)	115 600 (1.0)	5	266 (1.0)	499 660 (**0.79**)	144 400 (1.25)	4	180 (**0.68**)	2792
AL3	1 427 958 (1.0)	318 096 (1.0)	4	7758 (1.0)	1 256 812 (**0.88**)	336 400 (1.06)	4	5840 (**0.75**)	1906
AL4	1 938 427 (1.0)	366 025 (1.0)	5	18 312 (1.0)	1 462 919 (**0.75**)	384 400 (1.05)	4	3627 (**0.20**)	4857
AL5	9 169 149 (1.0)	1 690 000 (1.0)	5	145 466 (1.0)	9 065 222 (**0.99**)	1 960 000 (1.16)	5	64 023 (**0.44**)	4368
AL6	12 961 514 (1.0)	2 220 100 (1.0)	5	159 505 (1.0)	11 060 988 (**0.85**)	2 280 100 (1.03)	5	102 472 (**0.64**)	10 859
AL7	22 742 642 (1.0)	3 422 500 (1.0)	5	203 361 (1.0)	18 873 297 (**0.83**)	3 610 000 (1.05)	5	108 398 (**0.53**)	11 590
MP1	243 425 (1.0)	52 900 (1.0)	4	208 (1.0)	218 719 (**0.90**)	57 600 (1.09)	4	69 (**0.33**)	801
MP2	320 402 (1.0)	62 500 (1.0)	4	175 (1.0)	310 355 (**0.97**)	78 400 (1.25)	4	100 (**0.57**)	939
MP3	1 863 999 (1.0)	354 025 (1.0)	4	6742 (1.0)	1 722 149 (**0.92**)	360 000 (1.02)	4	2510 (**0.37**)	3084
MP4	1 716 339 (1.0)	374 544 (1.0)	5	4634 (1.0)	1 702 177 (**0.99**)	384 400 (1.03)	4	1523 (**0.33**)	1681
MP5	2727 331 (1.0)	659 344 (1.0)	5	17 655 (1.0)	2 729 148 (1.00)	672 400 (1.02)	4	3676 (**0.21**)	659

TABLE IV

COMPARISON OF WIRELENGTH, DIE AREA, RUNTIME, AND THE NUMBER OF TSVS WHEN THE DIE COUNT VARIES. THE TSV CO-PLACEMENT SCHEME AND 1× TSVS ARE USED. 3-D-**n** DENOTES *n*-DIE IMPLEMENTATION. ALL THE NUMBERS EXCEPT # TSVS ARE SCALED TO 2-D IMPLEMENTATION

ckt	Wirelength			Area			Runtime			# TSVs		
	3-D-2	3-D-3	3-D-4	3-D-2	3-D-3	3-D-4	3-D-2	3-D-3	3-D-4	3-D-2	3-D-3	3-D-4
AL1	0.96	1.01	0.95	1.06	1.20	1.29	0.69	0.41	0.36	337	1, 265	1, 305
AL2	0.99	0.82	0.79	1.07	1.18	1.25	1.01	0.78	0.68	1, 035	1, 945	2, 792
AL3	0.93	0.94	0.88	1.01	1.05	1.06	0.80	0.64	0.75	675	1, 902	1, 906
AL4	0.88	0.83	0.75	1.02	1.05	1.05	0.31	0.28	0.20	1, 745	3, 713	4, 857
AL5	0.98	0.96	0.99	1.14	1.14	1.16	0.42	0.47	0.44	1, 559	3, 664	4, 368
AL6	0.89	0.88	0.85	1.00	1.07	1.03	0.58	0.76	0.64	3, 838	8, 764	10, 859
AL7	0.92	0.87	0.83	1.01	1.03	1.05	0.68	0.62	0.53	4, 390	13, 154	11, 590
MP1	0.96	0.95	0.90	1.04	1.07	1.09	0.49	0.43	0.33	292	534	801
MP2	0.97	0.97	0.97	1.04	1.25	1.25	0.60	0.64	0.57	321	1, 044	939
MP3	1.02	0.93	0.92	1.00	1.01	1.02	0.68	0.86	0.37	1, 045	1, 542	3, 084
MP4	1.03	0.95	0.99	1.00	1.02	1.03	1.16	0.99	0.33	424	1, 056	1, 681
MP5	0.96	0.99	1.00	1.00	1.05	1.02	0.26	0.26	0.21	114	1, 706	659

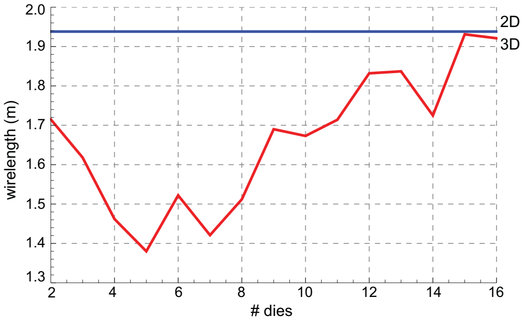


Fig. 13. Wirelength versus number of dies of AL4.

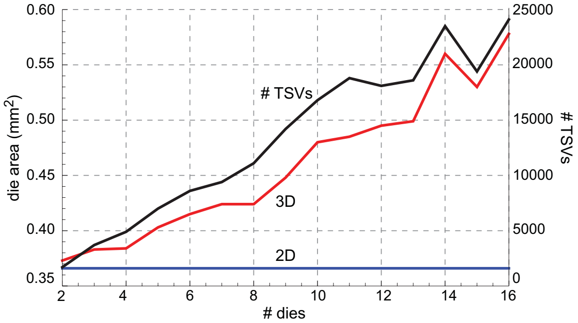


Fig. 14. Die area and number of TSVs versus number of dies of AL4.

chosen for each net sequentially in R-MST, so the additional runtime is very small even for big circuits such as AL6 or AL7. Sorting nets in R-MST takes $O(N \log N)$ where N is the total number of 3-D nets, Kruskal's algorithm takes $O(E_i \log E_i)$ where E_i is the total number of edges of net i, and selecting a TSV for an edge takes $O(T)$ where T is the total number of TSVs. Therefore, the complexity of R-MST is $O(N \cdot E_i \cdot T)$. Although it is a cubic algorithm, it is very fast in reality because the number of 3-D nets is not large and E_i is usually quite small (e.g., the former is less than 5000 and the latter is less than 500 in the simulation of this paper). As Table VI lists, the additional runtime for TSV assignment in R-MST is very small as compared to the runtime of 3-D placement.

The additional runtime for R-PL is also very small. In the second placement phase, many pins, which consist of I/O pins and placed cells, determine the location of each TSV in the

TABLE V

WIRELENGTH COMPARISON FOR TSV PLACEMENT TYPES (SCALED TO THE 2-D PLACEMENT RESULT). IR DENOTES TSV CO-PLACEMENT, R-MST IS MST-BASED TSV SITE PLACEMENT, R-PL IS PLACEMENT-BASED TSV SITE PLACEMENT, AND R-NS IS THE TSV SITE PLACEMENT SCHEME USING THE NEIGHBORHOOD SEARCH-BASED TSV ASSIGNMENT PRESENTED IN [29]. 1× TSVS ARE USED. 3-D-n DENOTES n-DIE IMPLEMENTATION

ckt	2-D	3-D-2					3-D-3					3-D-4				
		IR	R-MST	R-PL	R-NS	R-NF	IR	R-MST	R-PL	R-NS	R-NF	IR	R-MST	R-PL	R-NS	R-NF
AL1	257 051 μm (1.00)	0.96	1.02	1.04	1.05	1.00	1.01	1.08	1.11	1.10	1.07	0.95	1.05	1.05	1.07	1.04
AL2	631 818 μm (1.00)	0.99	1.10	1.12	1.14	1.07	0.82	0.95	0.96	0.95	0.95	0.79	0.93	0.92	0.92	–
AL3	1 427 958 μm (1.00)	0.93	1.00	1.00	0.99	0.98	0.94	0.98	1.03	1.02	0.96	0.88	0.95	0.96	0.96	0.94
AL4	1 938 427 μm (1.00)	0.88	0.93	0.94	0.94	0.93	0.83	0.95	0.99	0.96	–	0.75	0.82	0.85	0.83	–
AL5	9 169 149 μm (1.00)	0.98	1.01	1.02	1.03	1.00	0.96	1.05	1.05	1.04	–	0.99	1.03	1.03	1.03	–
AL6	12 961 514 μm (1.00)	0.89	0.93	0.96	0.95	–	0.88	0.97	1.01	1.00	–	0.85	0.95	0.99	0.98	–
AL7	22 742 642 μm (1.00)	0.92	0.96	0.97	0.97	–	0.87	0.96	0.99	0.98	–	0.83	0.88	0.91	0.92	–
MP1	243 425 μm (1.00)	0.96	1.05	1.05	1.06	1.04	0.95	1.01	1.06	1.04	1.00	0.90	0.98	0.98	0.97	0.97
MP2	320 402 μm (1.00)	0.97	1.00	1.00	1.00	1.00	0.97	1.02	1.05	1.02	1.02	0.97	1.02	1.03	1.03	1.03
MP3	1 863 999 μm (1.00)	1.02	1.04	1.04	1.04	1.04	0.93	0.98	0.99	0.99	0.97	0.92	1.01	1.02	1.01	–
MP4	1 716 339 μm (1.00)	1.03	1.07	1.08	1.08	1.07	0.95	1.00	1.01	1.01	0.99	0.99	1.06	1.06	1.06	1.04
MP5	2 727 331 μm (1.00)	0.96	1.01	1.01	1.02	1.00	0.99	1.04	1.04	1.04	1.04	1.00	1.05	1.05	1.07	1.05
Geomean	1.00	0.96	1.01	1.02	1.02		0.92	1.00	1.02	1.01		0.90	0.97	0.99	0.98	

TABLE VI

COMPARISON OF RUNTIME FOR TSV ASSIGNMENT FOR THE TWO TSV SITE PLACEMENT SCHEMES SHOWN IN TABLE V. IR AND R DENOTE RUNTIMES OF TSV CO-PLACEMENT AND TSV SITE PLACEMENT, RESPECTIVELY. A(R-MST) AND A(R-PL) DENOTE ADDITIONAL RUNTIMES FOR TSV ASSIGNMENT OF R-MST AND R-PL, RESPECTIVELY. THE UNIT IS SECOND (THE NUMBERS IN PARENTHESES DENOTE TOTAL ITERATION COUNTS OF MATRIX COMPUTATION DURING PLACEMENT)

ckt	3-D-2				3-D-3				3-D-4			
	Placement		Assignment		Placement		Assignment		Placement		Assignment	
	IR	R	A(R-MST)	A(R-PL)	IR	R	A(R-MST)	A(R-PL)	IR	R	A(R-MST)	A(R-PL)
AL1	98	77	0.14	0.18 (0)	58	72	0.28	0.71 (0)	51	70	0.25	0.77 (0)
AL2	267	260	0.30	0.51 (0)	207	208	0.52	1.28 (0)	180	195	0.70	2.11 (0)
AL3	6, 229	5, 691	0.69	0.81 (0)	4, 983	3, 577	1.05	1.33 (0)	5, 840	4, 950	0.99	0.70 (0)
AL4	5, 622	5, 287	0.34	2.77 (0)	5, 078	4, 106	0.98	2.94 (0)	3, 627	3, 911	1.37	4.76 (0)
AL5	61, 113	63 034	5.02	19.25 (0)	68 388	55 658	8.55	49.41 (1)	64 023	47 863	10.53	63.21 (2)
AL6	92 865	137 784	23.81	8.59 (0)	121 686	125 777	36.32	9.33 (0)	102 472	101 031	40.32	13.51 (0)
AL7	139 077	141 031	51.30	13.87 (0)	126 811	107 942	136.59	18.85 (0)	108 398	95 871	95.71	17.08 (0)
MP1	102	119	0.97	0.89 (1)	90	109	0.95	0.46 (0)	69	75	0.97	0.61 (0)
MP2	105	114	0.25	0.21 (0)	112	102	0.37	0.50 (0)	100	82	0.30	0.44 (0)
MP3	4613	5932	0.84	1.32 (0)	5835	4943	0.49	1.92 (0)	2510	4925	1.37	2.35 (0)
MP4	5354	5130	0.10	0.93 (0)	4569	4023	0.23	1.21 (0)	1523	2201	0.41	1.78 (0)
MP5	4551	6312	0.15	1.04 (0)	4551	5298	0.57	1.87 (0)	3676	3563	0.22	2.00 (0)

initial placement, so little overlap exists between TSVs. Thus, the placement in R-PL needs only a few iterations of matrix computation. The numbers in parentheses in Table VI show the number of iterations. Almost all of them are zero except a few cases in which only one or two iterations are necessary to remove all the overlaps among TSVs and cells.

On the other hand, the runtime for R-NS is almost negligible and the runtime for R-NF is prohibitively high. Therefore, the runtime for R-NS and R-NF is not shown in Table VI.

E. Impact of TSV Size

Using large TSVs results in large die area overhead, thereby degrading total wirelength. It also causes more serious overlap among TSVs and cells, thereby increasing runtime for 3-D placement. Therefore, the impact of TSV size on wirelength, die area, and runtime is investigated in this simulation.

Table VII lists the results for 1×, 2×, and 3× TSV size. Wirelength always increases as the TSV size increases, as does the die area. When the TSV becomes 2× larger, the wirelength increases by 9% on average and the die area increases by 27% on average. However, the wirelength becomes 28% longer and the die area becomes 74% larger on average when the TSV becomes 3× larger. The runtime increase is also not negligible. Therefore, the use of larger TSVs causes serious wirelength, area, and runtime overhead.

TABLE VII

COMPARISON OF WIRELENGTH, DIE AREA, AND RUNTIME WHEN THE TSV SIZE VARIES. THE TSV CO-PLACEMENT SCHEME IS USED WITH FOUR DIES (SCALED TO THE 1 × TSV CASE)

ckt	Wirelength			Area			Runtime		
	1×	2×	3×	1×	2×	3×	1×	2×	3×
AL1	1.00	1.04	1.68	1.00	1.46	2.20	1.00	1.87	2.21
AL2	1.00	1.14	1.37	1.00	1.68	2.74	1.00	1.63	1.95
AL3	1.00	1.08	1.32	1.00	1.16	1.72	1.00	1.55	1.80
AL4	1.00	1.18	1.36	1.00	1.31	2.05	1.00	1.32	1.30
AL5	1.00	1.07	1.12	1.00	1.12	1.30	1.00	1.44	1.62
AL6	1.00	1.11	1.14	1.00	1.35	1.58	1.00	1.34	1.77
AL7	1.00	1.07	1.10	1.00	1.19	1.42	1.00	1.36	1.41
MP1	1.00	1.20	1.75	1.00	1.38	2.01	1.00	1.38	2.11
MP2	1.00	1.06	1.34	1.00	1.37	1.92	1.00	1.50	1.86
MP3	1.00	1.07	1.21	1.00	1.20	1.70	1.00	2.11	3.59
MP4	1.00	1.03	1.09	1.00	1.11	1.40	1.00	1.52	2.30
MP5	1.00	1.01	1.05	1.00	1.08	1.36	1.00	1.03	1.14
Geomean	1.00	1.09	1.28	1.00	1.27	1.74	1.00	1.48	1.84

TABLE VIII
WIRELENGTH COMPARISON ($\times 10^5$ μm) WITH [15] AND [17]

ckt	[15]	[17]	This paper	ckt	[15]	[17]	This paper
AL1	2.84	2.59	2.44	MP1	3.01	2.39	2.19
AL2	5.83	5.52	5.00	MP2	3.88	3.87	3.10
AL3	14.01	13.51	12.57	MP3	20.02	17.87	17.22
AL4	19.12	22.91	14.63	MP4	26.83	18.27	17.02
				MP5	39.46	29.80	27.29
Avg.	1.00	1.01	0.84		1.00	0.84	0.74

VII. CONCLUSION

In this paper, the impact of TSVs on the 3-D stacked IC layout has been investigated. First, design issues newly introduced in 3-D ICs were discussed, and then two 3-D IC design flows were proposed. In the TSV co-placement scheme, gates and TSVs were placed simultaneously, whereas in the TSV site scheme, TSVs were uniformly placed and then gates were placed. The simulation results showed that 3-D designs have shorter wirelength, require fewer metal layers for routing, and shorter runtime for placement. However, die area increases because of TSV insertion. In conclusion, 3-D IC design methodologies and algorithms should take TSV placement and routing into account, which our 3-D IC design methodologies and algorithms perform effectively and efficiently.

APPENDIX

The 3-D global placer presented in [15] has been improved for this paper. Table VIII compares wirelength of [15], [17], and this paper.

REFERENCES

[1] J. W. Joyner, P. Zarkesh-Ha, J. A. Davis, and J. D. Meindl, "A three-dimensional stochastic wire-length distribution for variable separation of strata," in *Proc. IEEE Int. Interconnect Technol. Conf.*, Jun. 2000, pp. 126–128.

[2] D. H. Kim, S. Mukhopadhyay, and S. K. Lim, "Through-silicon-via aware interconnect prediction and optimization for 3D stacked ICs," in *Proc. ACM/IEEE Int. Workshop Syst. Level Interconnect Predict.*, Jul. 2009, pp. 85–92.

[3] T. Thorolfsson, K. Gonsalves, and P. D. Franzon, "Design automation for a 3DIC FFT processor for synthetic aperture radar: A case study," in *Proc. ACM Design Autom. Conf.*, Jul. 2009, pp. 51–56.

[4] D. H. Kim and S. K. Lim, "Through-silicon-via-aware delay and power prediction model for buffered interconnects in 3D ICs," in *Proc. ACM/IEEE Int. Workshop Syst. Level Interconnect Predict.*, Jun. 2010, pp. 25–32.

[5] X. Dong and Y. Xie, "System-level cost analysis and design exploration for three-dimensional integrated circuits (3D ICs)," in *Proc. Asia South Pacific Design Autom. Conf.*, Jan. 2009, pp. 234–241.

[6] K. Bernstein, P. Andry, J. Cann, P. Emma, D. Greenberg, W. Haensch, M. Ignatowski, S. Koester, J. Magerlein, and R. Puri, "Interconnects in the third dimension: Design challenges for 3D ICs," in *Proc. ACM Design Autom. Conf.*, Jun. 2007, pp. 562–567.

[7] E. Beyne, P. D. Moor, W. Ruythooren, R. Labie, A. Jourdain, H. Tilmans, D. S. Tezcan, P. Soussan, B. Swinnen, and R. Cartuyvels, "Through-silicon via and die stacking technologies for microsystems-integration," in *Proc. IEEE IEDM*, Dec. 2008, pp. 1–4.

[8] M. Koyanagi, T. Fukushima, and T. Tanaka, "High-density through silicon vias for 3-D LSIs," *Proc. IEEE*, vol. 97, no. 1, pp. 49–59, Jan. 2009.

[9] H. Chaabouni, M. Rousseau, P. Leduc, A. Farcy, R. El Farhane, A. Thuaire, G. Haury, A. Valentian, G. Billiot, M. Assous, F. De Crecy, J. Cluzel, A. Toffoli, D. Bouchu, L. Cadix, T. Lacrevaz, P. Ancey, N. Sillon, and B. Flechet, "Investigation on TSV impact on 65 nm CMOS devices and circuits," in *Proc. IEEE IEDM*, Dec. 2010, pp. 35–38.

[10] H. Y. Li, E. Liao, X. F. Pang, H. Yu, X. X. Yu, and J. Y. Sun, "Fast electroplating TSV process development for the via-last approach," in *Proc. IEEE Electron. Comp. Technol. Conf.*, Jun. 2010, pp. 777–780.

[11] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-aware 3D IC placement via transformation," in *Proc. Asia South Pacific Design Autom. Conf.*, Jan. 2007, pp. 780–785.

[12] B. Goplen and S. Sapatnekar, "Placement of 3D ICs with thermal and interlayer via considerations," in *Proc. ACM Design Autom. Conf.*, Jun. 2007, pp. 626–631.

[13] B. Goplen and S. Sapatnekar, "Efficient thermal placement of standard cells in 3D ICs using a force directed approach," in *Proc. IEEE Int. Conf. Comput.-Aided Design*, Nov. 2003, pp. 86–89.

[14] J. Cong and G. Luo, "A multilevel analytical placement for 3D ICs," in *Proc. Asia South Pacific Design Autom. Conf.*, Jan. 2009, pp. 361–366.

[15] D. H. Kim, K. Athikulwongse, and S. K. Lim, "A study of through-silicon-via impact on the 3D stacked IC layout," in *Proc. IEEE Int. Conf. Comput.-Aided Design*, Nov. 2009, pp. 674–680.

[16] M. Pathak, Y.-J. Lee, T. Moon, and S. K. Lim, "Through-silicon-via management during 3D physical design: When to add and how many?" in *Proc. IEEE Int. Conf. Comput.-Aided Design*, Nov. 2010, pp. 387–394.

[17] M.-K. Hsu, Y.-W. Chang, and V. Balabanov, "TSV-aware analytical placement for 3D IC designs," in *Proc. ACM Design Autom. Conf.*, Jun. 2011, pp. 664–669.

[18] G. Karypis and V. Kumar. *hMETIS, a Hypergraph Partitioning Package Version 1.5.3* [Online]. Available: http://glaros.dtc.umn.edu/gkhome/metis/hmetis/download

[19] *IWLS 2005 Benchmarks*. (2005) [Online]. Available: http://www.iwls.org/iwls2005

[20] *FreePDK45*. NCSU, Raleigh, NC [Online]. Available: http://www.eda.ncsu.edu/wiki/FreePDK

[21] *Nangate 45 nm Open Cell Library*. Nangate, Sunnyvale, CA [Online]. Available: http://www.nangate.com

[22] A.-C. Hsieh, T. Hwang, M.-T. Chang, M.-H. Tsai, C.-M. Tseng, and H.-C. Li, "TSV redundancy: Architecture and design issues in 3D IC," in *Proc. Design, Autom. Test Eur.*, Mar. 2010, pp. 166–171.

[23] M. Jung, J. Mitra, D. Z. Pan, and S. K. Lim, "TSV stress-aware full-chip mechanical reliability analysis and optimization for 3D IC," in *Proc. ACM Design Autom. Conf.*, Jun. 2011, pp. 188–193.

[24] D. H. Kim, R. O. Topaloglu, and S. K. Lim, "TSV density-driven global placement for 3D stacked ICs," in *Proc. Int. SoC Design Conf.*, Nov. 2011, pp. 135–138.

[25] *FaStack*. Tezzaron, Naperville, IL [Online]. Available: http://www.tezzaron.com

[26] J. Cong and S. K. Lim, "Edge separability based circuit clustering with application to multi-level circuit partitioning," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 23, no. 3, pp. 346–357, Mar. 2004.

[27] *Soc Encounter*. Cadence Design Systems, San Jose, CA [Online]. Available: http://www.cadence.com

[28] P. Spindler, U. Schlichtmann, and F. M. Johannes, "Kraftwerk2–a fast force-directed quadratic placement approach using an accurate net model," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 8, pp. 1398–1411, Aug. 2008.

[29] H. Yan, Z. Li, Q. Zhou, and X. Hong, "Via assignment algorithm for hierarchical 3-D placement," in *Proc. IEEE Int. Conf. Commun., Circuits Syst.*, May 2005, pp. 1225–1229.

[30] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logist.*, vol. 2, pp. 83–97, Mar. 1955.

[31] M.-C. Tsai, T.-C. Wang, and T. Hwang, "Through-silicon via planning in 3-D floorplanning," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 8, pp. 1448–1457, Aug. 2011.

Dae Hyun Kim (S'08) received the B.S. degree in electrical engineering from Seoul National University, Seoul, Korea, in 2002, and the M.S. degree in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, in 2007, where he is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering.

His current research interests include physical design algorithms particularly for 3-D integrated circuits, design methodology, placement, routing, and design for manufacturability.

Krit Athikulwongse (S'04) received the B.Eng., M.Eng., and M.S. degrees from the Department of Electrical Engineering, Chulalongkorn University, Bangkok, Thailand, in 1995, 1997, and 2005, respectively. He is currently pursuing the Ph.D. degree with the School of Electrical and Computer Engineering, Georgia Institute of Technology (Georgia Tech), Atlanta.

He joined the Computer Aided Design Laboratory, Georgia Tech, in 2008. His current research interests include physical design for 3-D integrated circuits.

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

He joined the School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, in 2001, where he is currently an Associate Professor. He has been leading the Cross-Center Theme on 3-D Integration for the Focus Center Research Program, Semiconductor Research Corporation, Durham, NC, since 2009. His current research interests include architectures, circuits, physical design for 3-D integrated circuits, and 3-D system-in-packages. He is the author of *Practical Problems in VLSI Physical Design Automation* (Springer, 2008).

Dr. Lim was a recipient of the Design Automation Conference Graduate Scholarship in 2003, the National Science Foundation Faculty Early Career Development CAREER Award in 2006, and the ACM SIGDA Distinguished Service Award in 2008. He was on the Advisory Board of the ACM Special Interest Group on Design Automation (SIGDA) from 2003 to 2008. He was an Associate Editor of the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS from 2007 to 2009. His papers were nominated for the Best Paper Award at the International Symposium on Physical Design in 2006, the International Conference on Computer-Aided Design in 2009, the Custom Integrated Circuits Conference in 2010, and the Design Automation Conference in 2011 and 2012. He was a member of the Design International Technology Working Group for the Renewal of the International Technology Roadmap for Semiconductors in 2009.