Received 31 October 2016; revised 31 October 2016; accepted 11 November 2016.
Date of publication 18 November 2016; date of current version 5 June 2019.

Digital Object Identifier 10.1109/TETC.2016.2630064

Wire Length Characteristics of Multi-Tier Gate-Level Monolithic 3D ICs

SHENG-EN DAVID LIN, (Student Member, IEEE) AND DAE HYUN KIM, (Member, IEEE)

The authors are with the School of Electrical Engineering and Computer Science, Washington State University, Pullman WA 99164
CORRESPONDING AUTHOR: D. H. KIM (daehyun@eecs.wsu.edu)

ABSTRACT Monolithic inter-layer vias (MIVs) used in 3D integration technology are much smaller than through-silicon vias (TSVs) used in TSV-based 3D integration. Thus, monolithic 3D integration provides shorter wire length, better performance, and lower power consumption than TSV-based 3D integration. Multi-tier monolithic 3D integration stacks more than two silicon layers, thereby increasing the device density, which reduces the wire length and power consumption and improves performance further than two-tier monolithic 3D integration. Since the size of an MIV is comparable to that of a local via, the wire lengths of multi-tier gate-level monolithic 3D IC layouts are expected to be close to the ideal wire length reduction ratio, for example, 29, 42, and 50 percent for two-, three-, and four-tier designs, respectively. However, it is still unknown whether it would be possible to achieve the ideal wire length reduction ratios by multi-tier gate-level monolithic 3D integration. In this paper, we propose an efficient routing methodology to design multi-tier monolithic 3D ICs. Using the design methodology, we design two- to eight-tier gate-level monolithic 3D IC layouts and thoroughly investigate the wire length characteristics of the layouts. The simulation results show that the proposed routing methodology reduces the wire length by 12 to 53 percent compared to the 2D designs. In addition, the proposed routing methodology builds multi-tier monolithic 3D IC layouts more efficiently than a state-of-the-art methodology with comparable routed wire length.

INDEX TERMS 3D IC, monolithic, multi-tier, 3D routing

I. INTRODUCTION

Stacking multiple device layers in gate-level three-dimensional integrated circuits (3D ICs) increases the device density, thereby reducing the total wire length, improving performance, and reducing power consumption [1], [2]. Thus, researchers have investigated the impact of stacking multiple tiers (dies) on the quality of gate-level 3D ICs [3], [4]. Some papers on the design of through-silicon-via(TSV)-based gate-level 3D IC layouts show that ideally the wire length of a gate-level 2D IC layout is reduced by $1/\sqrt{k}$ when k tiers are stacked, but the actual amount of wire length reduction is less than $1/\sqrt{k}$ due to the non-optimality of the design tools used to design the layouts and the non-negligible TSV size [2], [3].

Monolithic 3D integration is an advanced integration technology similar to the TSV-based 3D integration [5]. Vertical vias, so-called monolithic inter-layer vias (MIVs), used in monolithic 3D integration are much smaller than TSVs as shown in Figure 1. In addition, the device layers in monolithic 3D ICs are very thin. Thus, monolithic 3D integration significantly reduces the area and parasitic capacitance overhead of TSV-based 3D integration [6]. Recently, multi-tier monolithic 3D integration has been proposed [7]. Similar to multi-tier TSV-based 3D ICs, multi-tier monolithic 3D ICs are expected to provide shorter wire length, improved performance, and lower power consumption than two-tier monolithic 3D ICs.

Multi-tier gate-level monolithic 3D ICs can be designed by the methodology proposed for two-tier monolithic 3D ICs [8], [9]. For instance, 3D placement of the instances in a given design can be performed by placing cells in an $L \times L(um^2)$ layout using a 2D commercial tool, downscaling their locations by a constant scaling ratio r less than 1 to fit the instances into an $rL \times rL(um^2)$ floorplan area, and partitioning the instances into multiple tiers. Assuming the total silicon area is preserved, r is set to $1/\sqrt{k}$ where k is the number of tiers.

3D routing of multi-tier monolithic 3D ICs is a critical design step as finding good 3D topologies plays a big role in determining the quality of the designs. However, it is very

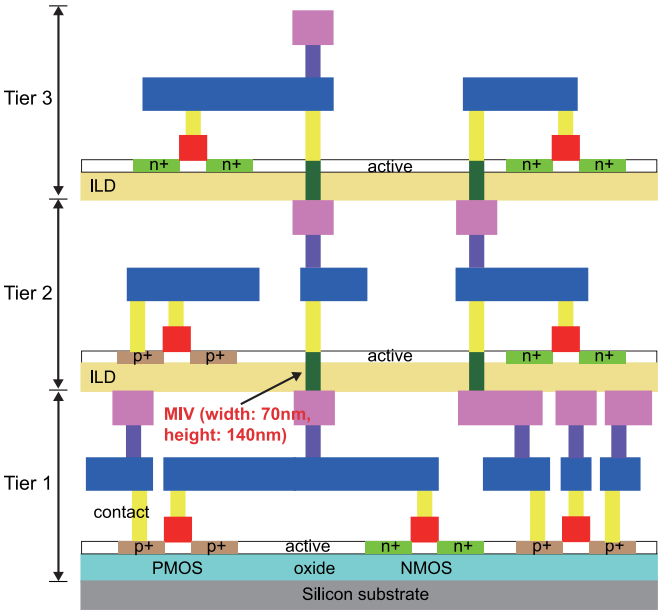

FIGURE 1. **Multi-tier monolithic 3D integration.**

challenging because 3D routing requires handling of many routing layers and numerous obstacles. For instance, routing of a four-tier monolithic 3D IC layout should be able to handle 24 routing layers when each tier has six routing layers. In addition, routing of 3D nets should insert MIVs into white-space while avoiding overlaps among MIVs and instances. Since stacking more tiers and distributing instances across the multiple silicon layers increases the number of MIVs, the runtime of routing multi-tier monolithic 3D ICs might also increase significantly as more tiers are stacked.

In this paper, we present an algorithm to route multi-tier gate-level monolithic 3D ICs, design several multi-tier monolithic 3D IC layouts, and investigate their wire length characteristics. This is the first work of using a 3D obstacle-avoiding routing algorithm to design multi-tier monolithic 3D ICs. Our contribution is as follows:

- We propose a routing methodology to route multi-tier gate-level monolithic 3D ICs.
- We design various multi-tier gate-level monolithic 3D ICs and thoroughly analyze and investigate the wire length characteristics of the designs.

The rest of this paper is organized as follows. In Section II, we review placement and routing methodologies developed to design gate-level monolithic 3D ICs and discuss the routing complexity of monolithic 3D ICs with some preliminary results. In Section III, we describe our routing methodology proposed to design multi-tier gate-level monolithic 3D ICs, complexity analysis of the algorithm, and a technique to reduce the runtime. Section IV shows simulation results and detailed analysis of the results, and we conclude in Section V.

II. PRELIMINARIES

In this section, we review the design methodologies proposed in the literature to build gate-level monolithic 3D ICs.

A. DESIGN METHODOLOGIES FOR TWO-TIER GATE-LEVEL MONOLITHIC 3D ICS

A few methodologies exist for two-tier gate-level monolithic 3D IC design [8]–[10]. Basically, all the design methodologies proposed in academia try to maximize the usage of commercial tools to build GDSII-level monolithic 3D IC layouts so that they can design high-quality layouts with a sufficiently-high accuracy in a reasonable amount of time.

Panth *et al.* proposed to use technology scaling to place instances in [8], [10]. For example, if a given circuit is designed in k tiers, the shapes (width, height, pin locations, etc.) of all the standard cells in the standard cell library are downscaled by a constant scaling ratio (typically $1/\sqrt{k}$). The floorplan area is scaled down by $1/k$ and all the instances in a given netlist are placed in the reduced floorplan area by a commercial tool with the modified standard cell library. After placement, there exist many instance overlaps because the 2D utilization increases. Thus, partitioning is performed to move the instances across the tiers to remove the overlaps and satisfy the density constraint for each bin of each tier. Routing is also performed by modifying the physical library file (LEF) as follows. First, the metal layers of the top tier are added to the routing layers of the physical library file. Then, the definition of each standard cell is duplicated, but the duplicated cells use the new metal layers for their pins and obstructions. Therefore, if two instances of the same logical type are placed at the same x, y location, but one of them uses the original definition and the other uses the duplicated definition, their pins do not overlap physically. Some commercial routers accept this situation and perform routing without any problem, so [8], [10] route two-tier designs in this way.

Lin *et al.* proposed to place instances in 3D using a commercial tool without library modification in [9]. Instead, they place instances in 2D and then downscale the locations of the instances by a scaling ratio $1/\sqrt{k}$ if it is designed in k tiers. In other words, if the location of an instance in the 2D placement result is (x, y), its location is moved to $(x/\sqrt{k}, y/\sqrt{k})$. Similarly, the floorplan area of the layout is also downscaled by $1/k$. Downscaling of the instance locations causes overlaps among the instances, so [9] also uses partitioning to move instances across the tiers.

B. DESIGN METHODOLOGIES FOR MULTI-TIER GATE-LEVEL MONOLITHIC 3D ICS

3D placement for multi-tier gate-level monolithic 3D ICs can be performed just by setting k to the number of tiers we want to design and performing k-way partitioning to move the instances to different tiers. For the 3D routing, we can add more routing layers to the physical library file and duplicate the cell definitions, i.e., create k definitions for each cell and the jth definition uses the routing layers defined for the jth tier. For example, if k is 3, three cell definitions are created for a cell and each of them uses its own metal layers for its pins and obstructions as shown in Figure 2.

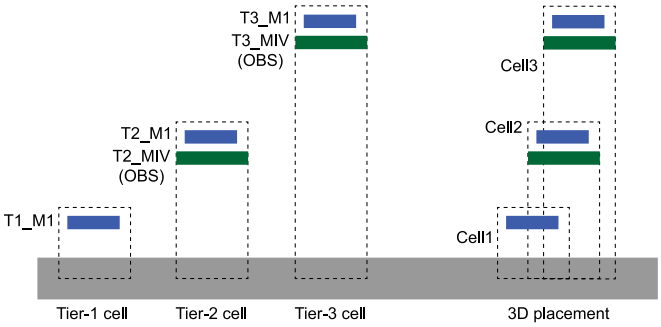

FIGURE 2. Duplicated standard cell definitions for 3D routing of three-tier designs. "OBS" means obstructions.

However, there is a critical problem in this routing methodology, which is that the routing complexity significantly goes up when more tiers are stacked. All the instances not in the bottommost tier are treated as routing obstacles during routing, so stacking more tiers leads to more routing obstacles. Thus, the routing runtime increases as more tiers are stacked.

Table 1 shows a preliminary result for four benchmark circuits. To generate the two-tier 3D IC layouts, we used the placement methodology developed in [9] with $k = 2$, the routing methodology developed in [8], and Cadence Encounter for 2D placement and 3D routing. In the table, we observe that the 3D to 2D wire length ratio is much greater than the ideal ratio, $1/\sqrt{k} \approx 0.71$ in some designs such as Ckt1 and Ckt4. In addition, the routing runtime for the two-tier designs is 12× to 22× longer than that for the 2D designs. This runtime overhead increases almost exponentially and routing a three-tier design takes almost 100× longer runtime than routing a 2D design based on our experience.

In this work, we propose a new routing methodology to route each tier separately in the design of multi-tier gate-level monolithic 3D ICs. Using the methodology and some commercial tools, we design several multi-tier gate-level monolithic 3D IC layouts and thoroughly investigate the wire length characteristics of the multi-tier monolithic 3D IC layouts.

III. ROUTING OF MULTI-TIER GATE-LEVEL MONOLITHIC 3D ICS

In this section, we describe a routing methodology for the design of multi-tier gate-level monolithic 3D ICs. The goal of 3D routing in this paper is to generate high-quality detailed routing results to accurately investigate the wire length characteristics of multi-tier gate-level monolithic 3D ICs. 3D routing can perform 3D global and detailed routing as the traditional 2D routing performs 2D global and detailed routing. However, we choose a different approach based on MIV insertion and 2D routing for the following reasons. First, MIV insertion removes the need for 3D global and detailed routing. Second, once MIV locations are found, each tier can be routed independently, which could reduce the routing runtime significantly. Third, this approach can generate GDSII-level layouts with the aid of commercial routers, which enables very accurate analysis of the quality of multi-tier gate-level monolithic 3D IC layouts. Thus, we first find MIV locations for all 3D nets, dump the MIV locations into design files, and then run 2D routing for each tier using a commercial tool.

TABLE 1. Normalized Wire Length (WL) and Runtime (T) for Routing of 2D and Two-Tier Monolithic 3D IC Layouts.

Circuit	2D		Two-tier 3D	
	WL	T	WL	T
Ckt1	1.00	1.00	0.78	12.70
Ckt2	1.00	1.00	0.77	22.17
Ckt3	1.00	1.00	0.74	12.02
Ckt4	1.00	1.00	0.86	19.22

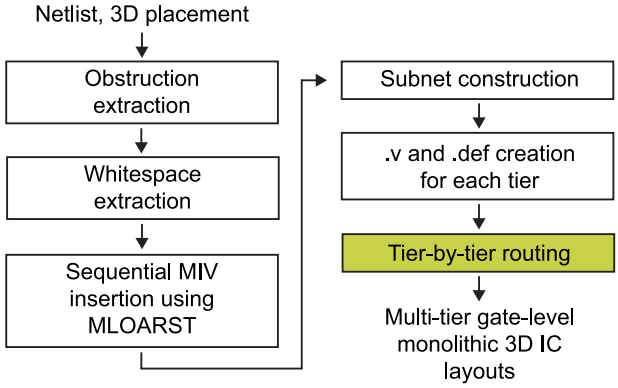

FIGURE 3. The proposed routing methodology. The colored step uses a commercial tool.

A. 3D ROUTING METHODOLOGY

Figure 3 shows an overview of the proposed methodology for routing of signal nets in the design of multi-tier gate-level monolithic 3D ICs. The details of the routing methodology are explained below:

- *Obstruction extraction:* For a given netlist and a 3D placement result, we first extract obstructions (instances and placement blockages) in each tier and store the information into an obstruction data structure. This obstruction information is used in multi-layer obstacle-avoiding rectilinear Steiner tree (MLOARST) construction.
- *Whitespace extraction:* We also extract whitespace available for MIV insertion and store the information into a whitespace data structure. We extract whitespace blocks and a whitespace block is defined as whitespace between two horizontally-adjacent instances.
- *Sequential MIV insertion:* We construct a 3D routing topology for each 3D net using an MLOARST construction algorithm [11] and extract MIV locations from the topology. For each MIV location obtained from the MLOARST, we find the nearest whitespace block and physically insert an MIV into the whitespace. If the utilization of the whitespace block reaches a pre-

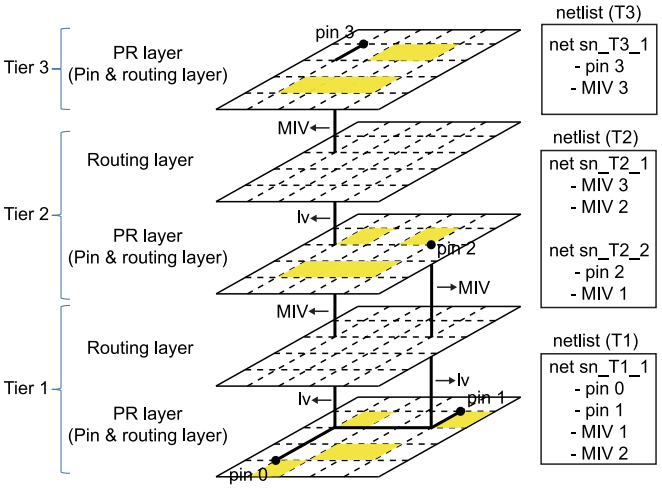

FIGURE 4. MIV insertion using MLOARST construction. lv denotes a local via. The colored parallelograms are obstructions such as instances.

determined number, we mark the whitespace as "used" and do not use it anymore.

- *Subnet construction:* Once we find actual MIV locations for each 3D net, we decompose the MLOARST constructed for the 3D net into multiple subnets as shown in Figure 4. For instance, the net in the figure has a subnet composed of pin 3 and MIV 3 in Tier 3, two subnets (sn_T2_1 and sn_T2_2) in Tier 2, and a subnet composed of pin 0, pin 1, MIV 1, and MIV 2 in Tier 1. sn_T2_1 consists of MIV 3 and MIV 2 and sn_T2_2 consists of pin 2 and MIV 1. Although all these subnets belong to the same net, they will be routed separately in each tier as if they belong to different nets. For example, if we look into Tier 2, sn_T2_1 and sn_T2_2 are not connected, so they are considered different nets. Merging all the subnets in 3D generate a full topology for the net.
- *.v and .def creation:* We construct a Verilog netlist and a DEF file for each tier.
- *Tier-by-tier routing:* The actual 3D routing is performed by a commercial tool. Since each tier is independent of other tiers, each tier is loaded into a commercial tool and routed separately.

B. MIV INSERTION FOR 3D NETS

We find MIV locations for a 3D net using an MLOARST construction algorithm as follows. Without loss of generality, suppose a 3D net spans from tier 1 to tier p where p is greater than 1 and tier 1 is the bottommost tier as shown in Figure 4. We insert two layers (a PR layer containing pins and obstructions and a routing layer) for each tier from tier 1 to tier $p-1$. For the topmost tier, we insert only a PR layer, so there are total $2p-1$ layers when the net spans from tier 1 to tier p. Although we use only two layers (a PR layer and a routing layer) for each tier, we need more routing layers for each tier if additional routing blockages other than on-chip instances exist in the given

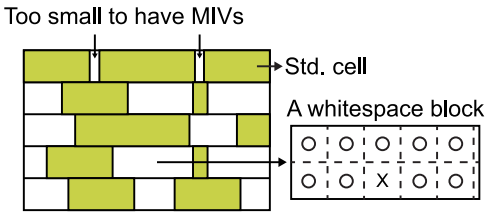

FIGURE 5. Whitespace management. A whitespace block is extracted from a given 3D layout. "O" and "X" denote available and unavailable whitespace for MIV insertion, respectively.

3D layout. Since we do not consider these blockages, we need only two routing layers per tier in this paper.

Next, we extract pin locations of the 3D net from a given 3D placement result and all obstructions inside the 3D bounding box of the 3D net. The pins and obstructions are placed in the PR layers as shown in Figure 4. Then, we use an MLOARST construction algorithm [11] to construct an MLOARST for the net. The vertical edges of the constructed MLOARST between a routing layer and a PR layer right above the routing layer are MIVs, whereas vertical edges between a PR layer and a routing layer right above the PR layer are local vias as shown in Figure 4. Thus, we obtain MIV locations from the locations of the vertical edges found in the MLOARST this way.

After we find MIV locations for the 3D net using MLOARST construction, we find a nearest whitespace block available for each MIV in the given 3D layout by searching nearby whitespace blocks extracted from the layout. Each whitespace block can accommodate multiple MIVs as shown in Figure 5, so we not only find a nearest available whitespace block, but also find the best available MIV location in the whitespace block. Once we find available whitespace, we insert an MIV into the location and mark it "used" (the "X" mark in the figure).

C. COMPLEXITY ANALYSIS

The complexity of the obstruction extraction step is $O(c)$ where c is the total number of instances. The complexity of the whitespace extraction step is also $O(c)$ because the number of the whitespace blocks is directly related to how many instances are placed in the design. Thus, geometrical sorting of the instances and row-by-row sweeping can extract whitespace, which takes $O(c)$ time. The complexity of an MLOARST construction is practically $O(v^2 \lg v)$ where v is the total number of pins and corner vertices as explained in [11].

We use the spiral search to find a nearest whitespace block available for each MIV found by MLOARST construction. If there is enough whitespace around the target MIV location, the complexity of the spiral search is $O(1)$. If the location of the nearest available whitespace is far away from the target MIV location, however, it takes $O(w)$ time to find a whitespace block where w is the number of whitespace blocks. Since there is a linear relationship between w and c, (i.e., $c \leq w \leq c+r$

TABLE 2. Comparison of the Wire Length and the Number of MIVs between Routing by a Commercial Tool (CT) and Routing by the Proposed Routing Methodology (Ours).

Circuit	# instances	# nets	Wire length (um)		# MIVs		Runtime (sec)	
			CT	Ours	CT	Ours	CT	Ours (Speed-up)
HCA-2T			281,561	280,408 (−0.4%)	5,994	5,725 (−4.5%)	764	68 (11.2X)
HCA-3T	17,412	19,461	260,190	271,573 (+4.4%)	10,407	10,458 (+0.5%)	1,659	160 (10.4X)
HCA-4T			253,760	276,784 (+9.1%)	14,380	14,572 (+1.3%)	2,557	233 (11.0X)
KSA-2T			587,939	608,246 (+3.5%)	12,363	11,990 (−3.0%)	1,596	179 (8.9X)
KSA-3T	29,732	32,283	583,920	631,901 (+7.5%)	21,347	21,698 (+1.6%)	3,422	736 (4.7X)
KSA-4T			617,973	771,618 (+24.9%)	30,150	31,424 (−4.2%)	7,975	1,307 (6.1X)
LDPC-2T			3,873,394	3,930,236 (+1.5%)	35,379	27,907 (−21.1%)	2,385	494 (4.8X)
LDPC-3T	50,753	54,785	3,208,639	3,314,321 (+3.3%)	50,103	45,398 (−9.4%)	3,738	620 (6.0X)
LDPC-4T			2,842,687	2,967,904 (+4.4%)	61,489	61,485 (−0.006%)	5,615	720 (7.8X)
DES-2T			936,172	875,242 (−6.5%)	21,089	27,114 (+28.6%)	1,110	909 (1.2X)
DES-3T	74,970	77,349	818,943	817,180 (−0.2%)	34,210	45,086 (+31.8%)	2,186	395 (5.5X)
DES-4T			845,370	803,267 (−5.0%)	46,805	61,762 (+32.0%)	3,371	233 (14.5X)
FFT-2T			6,158,711	6,057,097 (−1.6%)	84,693	98,275 (+16.0%)	6,655	1,130 (5.9X)
FFT-3T	255,710	259,427	5,779,923	5,680,158 (−1.7%)	125,705	147,513 (+17.3%)	19,400	1,226 (15.8X)
FFT-4T			5,639,077	5,325,941 (−5.6%)	163,438	206,040 (+26.1%)	26,007	1,140 (22.8X)

We compare them only up to four-tier designs (-kT in the table denotes k-tier designs).

where r is the number of standard cell rows in the layout), the complexity of the spiral search for an MIV is $O(c)$.

The number of MIVs for each 3D net varies depending on the topological distribution of the pins of the net. In general, however, the number of MIVs for a 3D net that connects m instances and spans from tier 1 to tier p is bounded above by $(p-1)\cdot(m-1)$, which is the number of MIVs when the snake-like routing occurs. Thus, the complexity of the MIV insertion for a 3D net is $O(m\cdot c)$ assuming p is small (p is also bounded above by the number of tiers in which the design is built). Thus, the complexity of the proposed routing methodology is $O(n\ m^2\ \lg\ m)$ where n is the total number of 3D nets and m is the average number of instances.

D. RUNTIME REDUCTION

The most time-consuming step in the proposed routing methodology is the MLOARST construction. Especially, if there are many obstructions in the 3D bounding box of a 3D net, the number of grids in the MLOARST construction increases significantly, so does the runtime. Thus, we propose a runtime reduction technique as follows. When the half-perimeter wirelength (HPWL) of the bounding box of a 3D net is shorter than a pre-determined number r_s, we take the obstructions inside the bounding box into account during MLOARST construction for the net, otherwise we ignore all the obstructions in the bounding box and construct an MLOARST. A rationale for the proposed runtime reduction technique is as follows. In general, whitespace is evenly distributed over the entire layout area, so even if we ignore the obstructions during MLOARST construction, it is highly likely that some whitespace exists around each target MIV location. In addition, if the bounding box of a 3D net is large, there will exist multiple optimal whitespace locations in the bounding box for each target MIV location. Thus, we construct an MLOARST without considering obstructions for a 3D net if the bounding box size of the net is greater than r_s. In the simulation section, we show the impact of r_s on the quality of 3D IC layouts.

IV. SIMULATION RESULTS

In this section, we present simulation settings, a multi-tier gate-level monolithic 3D IC design methodology we used, simulation results, and detailed analysis.

A. SIMULATION SETTINGS

We use the Nangate 45 nm standard cell library [12] for the physical design library, Synopsys Design Compiler for netlist synthesis, and Cadence Innovus for 2D placement, legalization, and tier-by-tier routing. Each tier has six metal layers and all the layers have the same width and pitch in the physical design library. An MIV is square-shaped, the width of an MIV is 70 nm, and the pitch of two adjacent MIVs is 140 nm. The keep-out zone for MIVs is defined as the minimum distance between the boundaries of an MIV and an adjacent instance and it is set to 70 nm. We use five benchmarks in this paper as shown in Table 2.

To design multi-tier gate-level monolithic 3D IC layouts, we use the design methodology in [9], which is shown in Figure 6. It first places instances on a 2D floorplan area using a commercial tool. Then, the location of each instance is downscaled by a constant ratio, i.e., the location (x_c, y_c) of instance c is changed to $(r\cdot x_c, r\cdot y_c)$ where r is the scaling ratio. Typically, r is set to $1/\sqrt{k}$ where k is the number of tiers. The floorplan area scales down by $1/k$. Since the downscaling of the instance locations generates instance overlaps, bin-by-bin k-way partitioning is performed to move some of the instances to other tiers while satisfying a given density constraint in each bin. In more detail, after downscaling the locations of the instances of a given 2D layout, we create a 3D grid and put the instances into the bins of the grid in the

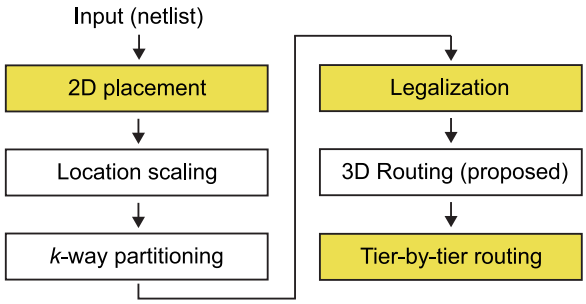

FIGURE 6. The design methodology used in this paper to build multi-tier gate-level monolithic 3D IC. Commercial tools are used in the colored steps.

bottommost tier. Then, we perform partitioning sequentially from bin 1 to bin B where B is the total number of bins in the bottommost tier. When we partition bin b into k partitions to place the instances in partition t into Tier t, we include the primary I/O pins and all the instances in bin 1 to bin $b-1$ that are connected to the instances in bin b to reduce the number of MIVs. Since the instances in bin $b+1$ to bin B have not been partitioned yet, we ignore them when we partition the instances in bin b. We use hMetis for k-way partitioning [13] and perform balanced partitioning so that all the instances are evenly distributed in all the x-, y-, and z-directions.

After partitioning, we perform legalization (overlap removal and instance snapping) using a commercial tool. Then, we use the routing methodology proposed in this paper to find MIV locations. The instances and MIVs information in each tier is dumped into a Verilog netlist and a DEF file and loaded into Cadence Innovus. Since each tier has its own netlist and DEF file, we perform routing for each tier separately.

B. THE QUALITY OF THE PROPOSED ROUTING METHODOLOGY

We first compare the quality of the proposed routing methodology with that of a commercial router for two- to four-tier gate-level monolithic 3D IC layouts to show that the proposed routing methodology produces high-quality 3D IC layouts. Table 2 compares the wire length and the number of MIVs of five two- to four-tier 3D IC layouts routed by a commercial tool [8] and by the proposed routing methodology (MIV insertion by the proposed algorithm with tier-by-tier routing using the same commercial tool). Routing by a commercial tool overall shows shorter wire length than the proposed routing methodology for the HCA, KSA, and LDPC designs. However, the proposed methodology achieves shorter wire length for the DES and FFT designs by 0.2 to 6.5 percent.

We also show the number of MIVs inserted by the commercial tool and the proposed routing methodology in Table 2. Since the primary objective function in routing is the planar wire length, we do not consider the number of MIVs during routing as long as there is a good use of them. However, inserting too many MIVs might cause routing congestion for 3D nets, so we assign a sufficiently large weighting factor to z-directional edges in the MLOARST construction so that the number of MIVs can be controlled. Both the commercial tool and the proposed methodology use similar amounts of MIVs for the HCA, KSA, and LDPC designs, but the commercial tool inserts fewer MIVs for the DES (-28 to -32 percent) and FFT (-16 to -26 percent) designs. The commercial tool might consider other factors during routing, so it is not fair to directly compare the wire length and the number of MIVs, but the table at least shows that the proposed routing methodology generates high-quality multi-tier monolithic 3D IC layouts.

The two rightmost columns in Table 2 also shows the runtime for routing. The runtime for routing using the commercial tool measures the total routing time (from the beginning of the global routing to the end of the detailed routing). The runtime using the proposed routing methodology is $t_{3D} + \mathrm{MAX}(t_{2D,1}, \ldots, t_{2D,k})$ where t_{3D} is the runtime from the obstruction extraction step to the design file generation step as shown in Figure 3 and $t_{2D,i}$ is the runtime for routing of 2D nets of ith tier using the commercial tool. As the table shows, the proposed routing methodology routes 3D designs much faster than the commercial tool by 1.2X to 22.8X.

C. WIRE LENGTH OF MULTI-TIER MONOLITHIC 3D IC DESIGNS

Table 3 shows the footprint area, the initial utilization, and the wire length of the 2D and multi-tier gate-level monolithic 3D IC designs. First of all, the ideal footprint area reduction ratio is $1/k$ where k is the number of tiers. Thus, when k is 2, 3, and 4, the ratio becomes 0.5, 0.33, and 0.25, respectively. However, the actual footprint area reduction ratios are greater than the ideal ratios because we do not shrink the power/ground ring area surrounding the core area.

TABLE 3. Footprint Area (FP), Initial Utilization (Util), and Wire Length (WL) of the 2D and Multi-Tier Gate-Level Monolithic 3D IC Designs.

Circuit	2D			3D (# tiers)													
				2		3		4		5		6		7		8	
	FP (um^2)	WL (um)	Util	FP	WL	FP	WL	FP	WL	FP	WL	FP	WL	FP	WL	FP	WL
HCA	48,839 (1.00)	319,495 (1.00)	0.60	0.59	**0.88**	0.44	**0.85**	0.36	**0.87**	-	-	-	-	-	-	-	-
KSA	74,954 (1.00)	720,643 (1.00)	0.60	0.57	**0.84**	0.42	**0.88**	0.34	**1.07**	-	-	-	-	-	-	-	-
LDPC	396,900 (1.00)	5,460,527 (1.00)	0.21	0.53	**0.72**	0.37	**0.61**	0.29	**0.54**	0.24	**0.50**	0.20	**0.48**	0.18	**0.47**	0.16	**0.47**
DES	278,783 (1.00)	1,094,604 (1.00)	0.60	0.51	**0.80**	0.35	**0.75**	0.26	**0.73**	-	-	-	-	-	-	-	-
FFT	1,111,964 (1.00)	6,715,956 (1.00)	0.60	0.51	**0.90**	0.34	**0.85**	0.26	**0.79**	-	-	-	-	-	-	-	-

We show the actual values only for the 2D designs and the ratios for 3D designs.

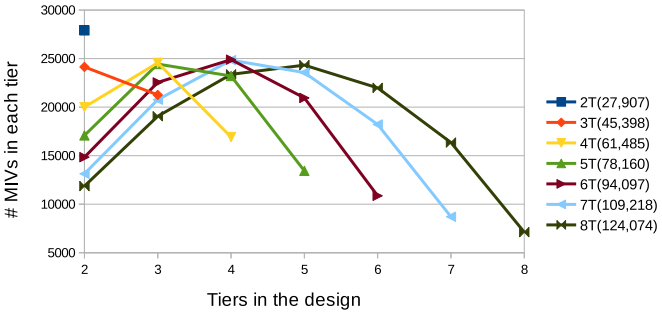

FIGURE 7. **# MIVs in each tier of the LDPC designs. kT is for the k-tier design. The numbers in the parentheses are the total number of MIVs in the designs.**

The wire length reduction ratio (WL_3D versus WL_2D) of the 3D designs generally increases as more tiers are stacked. However, the small designs (HCA and KSA) have the shortest wire length when two and three tiers are stacked. If more tiers are stacked for these designs, the wire length starts increasing. On the other hand, the wire length of the large designs (LDPC, DES, and FFT) keeps decreasing as more tiers are stacked. However, the decrement of the wire length reduction ratio also decreases in general and will eventually saturate if more tiers are stacked. If the tier count goes up beyond the saturation point, the wire length of the large designs will also go up. More details on the wire length saturation is explained in the next section.

Figure 7 shows that the tiers in the middle of the design have more MIVs and the tiers in both ends (especially, Tier 1 and Tier k) have fewer MIVs. In general, the routing resources in the middle of a layout are heavily used in 2D ICs. Similarly, the 3D routing resources (MIVs) in the middle tiers of a 3D IC layout are more heavily used than those in the bottommost and the topmost tiers. This explains why more MIVs are inserted into the middle tiers in Figure 7. Figure 8 shows the distribution of MIVs in Tier 2 to Tier 8 in the eight-tier LDPC design (Tier 1 does not have MIVs). Table 4 shows the total area used by the MIVs in the two- to eight-tier LDPC designs.

TABLE 4. **Total Silicon Area Used by MIVs in the LDPC Designs.**

Design	MIV area (um^2)	MIV area / Total silicon area
LDPC-2T	137	0.03%
LDPC-3T	222	0.05%
LDPC-4T	301	0.06%
LDPC-5T	383	0.08%
LDPC-6T	461	0.10%
LDPC-7T	535	0.11%
LDPC-8T	608	0.12%

D. SATURATION OF THE WIRE LENGTH REDUCTION

To analyze the saturation of the wire length reduction more accurately, we show the total number of MIVs and the number of MIVs in each tier of the LDPC designs in Figure 7. As the figure shows, the total number of MIVs goes up as more tiers are stacked. Since the total silicon area (the footprint area multiplied by the number of tiers) does not change, the MIV density goes up if we insert more MIVs into a design. Since an MIV is defined as an I/O pin in the Verilog netlist, increasing the MIV density also increases the total pin density in the design, which aggravates the routing congestion. Since higher routing congestion generally results in more routing detours, the wire length reduction ratio decreases as more tiers are stacked and the total wire length finally starts to increase. Table 5 shows the pin density and the wire length of each tier in the two- to eight-tier LDPC designs. As the table shows, the pin density goes up as more tiers are stacked because more MIVs are inserted into the reduced footprint area. For instance, for the eight-tier design (the last row in the table), the pin densities of the middle tiers increase significantly from 0.487 (Tier 1) to 0.983 (Tier 4), so the wire lengths of the middle tiers are in general longer than those of other tiers.

Figure 9 shows the variation of the length of a few randomly selected nets in the two- to eight-tier LDPC designs. Although we randomly chose the nets, most of the other nets

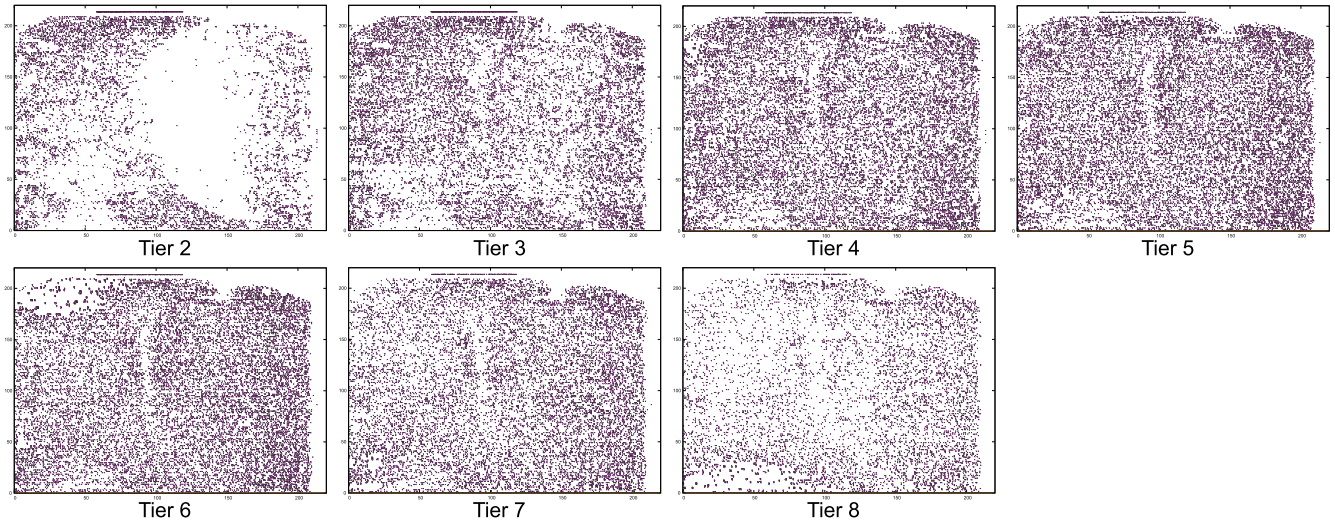

FIGURE 8. **MIV distribution in each tier of the eight-tier LDPC design.**

TABLE 5. The Pin Density (d) and the Wire Length (l) of Each Tier in the k-Tier LDPC Designs.

k		T1	T2	T3	T4	T5	T6	T7	T8
1	d	0.687	-	-	-	-	-	-	-
	l	5.46	-	-	-	-	-	-	-
2	d	0.620	0.804	-	-	-	-	-	-
	l	1.68	2.25	-	-	-	-	-	-
3	d	0.595	0.867	0.712	-	-	-	-	-
	l	0.85	1.38	1.08	-	-	-	-	-
4	d	0.567	0.874	0.833	0.669	-	-	-	-
	l	0.52	0.94	0.86	0.65	-	-	-	-
5	d	0.547	0.872	0.852	0.870	0.606	-	-	-
	l	0.38	0.70	0.62	0.63	0.42	-	-	-
6	d	0.531	0.860	0.865	0.930	0.857	0.536	-	-
	l	0.28	0.56	0.48	0.53	0.49	0.28	-	-
7	d	0.514	0.851	0.857	0.973	0.917	0.844	0.483	-
	l	0.23	0.45	0.39	0.46	0.43	0.39	0.20	-
8	d	0.487	0.846	0.859	0.983	0.955	0.932	0.830	0.432
	l	0.20	0.40	0.34	0.41	0.39	0.37	0.33	0.15

The pin density is measured by the number of pins (instance pins and MIVs) divided by the footprint area. Tn denotes Tier n. The unit of the wire length is meter.

show similar trends. The variation is measured by the deviation of the net length from its ideal value. In other words, the variation is defined as $(a-b)/b$ where a is the length of the net in the k-tier design and b is the expected ideal length of the net, which is defined as $l_{2D}/\sqrt{k}$ where l_{2D} is the length of the net in the original 2D design. Thus, if the variation of a net in the k-tier design is close to zero, the length of the net in the design is close to the ideal value, which is $l_{2D}/\sqrt{k}$. The variation could be negative if the length is shorter than its ideal value. As the figure shows, the deviation generally increases as more tiers are stacked. The increase of the deviation is due to higher routing congestion caused by inserting more MIVs as we stack more tiers. However, the deviation of a net sometimes decreases temporarily when the tier count increases. This is due to a few different reasons such as 1) the range[1] of the net might decrease when one more tier is stacked, so the net requires fewer MIVs, which leads to a shorter detour or 2) the refinement step in the legalization did not move the instances connected to the net, so the optimality of the net was not degraded by legalization.

E. ROUTING CONGESTION

As more tiers are stacked to design a multi-tier gate-level monolithic 3D IC layout, the total wire length generally decreases until it reaches a saturation point. However, the number of MIVs increases at the same time, so routing congestion is under question. In this section, therefore, we investigate routing congestion of multi-tier gate-level monolithic 3D IC designs. To quantify routing congestion, we split 3D layouts into bins of size 20 um by 20 um and compute the total wire area in each bin. Figure 10 shows a histogram where the x-axis shows the routing resource utilization (wire area * 100/ bin area) and the y-axis shows the number of bins belonging to the utilization. As the figure shows, the routing resource utilization generally decreases as more tiers are stacked. This is because the average instance-to-instance distance goes down when the tier count goes up, which leads to wire length reduction resulting in routing resource utilization reduction. Thus, we observe that routing congestion goes down as more tiers are stacked until the total wire length saturates.

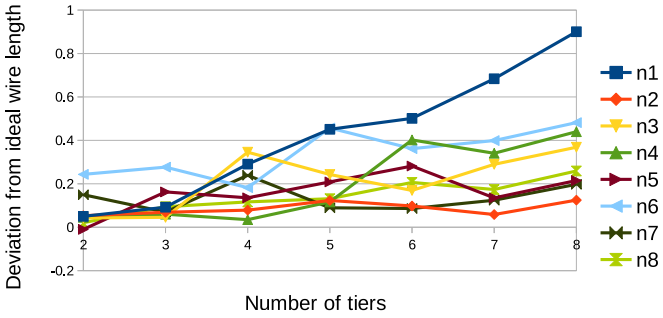

FIGURE 9. Variation of the length of a few randomly selected nets in the LDPC designs.

F. IMPACT OF THE RUNTIME REDUCTION TECHNIQUE

As explained in Section III-D, we ignore the obstructions in the 3D bounding box of a net if the HPWL of the net is greater than a pre-determined number (r_s). Routing a 3D net without considering the obstructions, however, might result in wire length overhead. Thus, we show the impact of r_s on the wire length in this experiment by building two-tier LDPC designs with r_s =10um and r_s =30 um and comparing their wire length and runtime. When r_s is 10 um, the total wire length is 3,930,236 um and the runtime is 494 seconds. On the other hand, when r_s is 30 um, the total wire length is 3,936,455 um and the runtime is 807 seconds. Increasing r_s is expected to generate higher-quality (shorter wire length) layouts, but require longer runtime for routing. However, we observe from our experiment that they have similar wire length results, but decreasing r_s reduces the runtime for routing.

G. IMPACT OF THE QUALITY OF 3D PLACEMENT ON 3D ROUTING

In this section, we investigate the impact of the quality of 3D placement results on 3D routing. We built four different 3D placement results for four-tier LDPC designs, measured their total 3D HPWL,[2] and compared the values with their routed wire length. Table 6 shows the result. As the 3D HPWL increases, the routed wire length also increases as shown in the table. However, the number of MIVs varies somewhat randomly. This is because the MIV count is related more closely to the cut size. Thus, even if 3D layout L_1 has shorter

[1]When a net spans from tier i to tier j ($i < j$), the *range* of the net is defined as $[i,j]$.

[2]The 3D HPWL of a net is computed by the sum of the HPWL of its subnets in each tier.

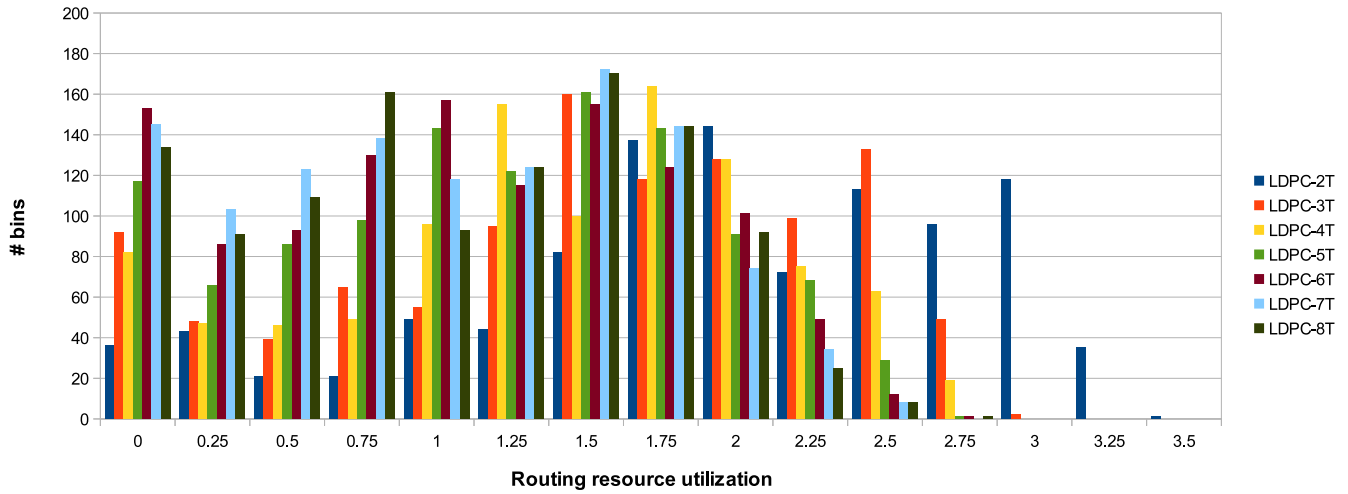

FIGURE 10. Quantification of routing congestion in multi-tier gate-level monolithic 3D IC layouts. The *x*-axis shows routing resource utilization, which is measured by $(A * 100)/B$ where A is the total wire area in each bin and B is the bin area. The *y*-axis shows the number of bins.

wire length than 3D layout L_2, L_1 might have more MIVs than L_2 if L_1 has more 3D nets than L_2 or if the total cut size of L_1 is greater than that of L_2.

V. CONCLUSION

In this paper, we proposed a new routing methodology to design multi-tier gate-level monolithic 3D ICs. We validated its quality by comparing the 3D IC layouts generated by the proposed methodology and a commercial routing tool. Using the routing methodology, we designed several multi-tier gate-level monolithic 3D IC layouts and analyzed their wire length characteristics. As more tiers are stacked, the wire length of each benchmark circuit decreases initially, and up to a certain number of tiers begins to saturate and starts to increase similarly to TSV-based 3D ICs. Next, we thoroughly investigated the saturation phenomena of the wire length reduction in multi-tier gate-level monolithic 3D ICs. In our analysis, we found that when a benchmark circuit is designed in k tiers, if k increases, we insert more MIVs. The increase of the number of MIVs leads to the increase of pin density in each tier, so the routing congestion goes up. When k is small, the pin density increase does not cause routing congestion and the footprint area goes down, so the total wire length generally decreases. When k goes up, however, the pin density causes routing congestion, so the total wire length eventually starts increasing. Thus, even if monolithic 3D integration is used to reduce the wire length of a circuit more effectively than TSV-based 3D integration, a wire length saturation point exists for each circuit.

TABLE 6. Impact of the Quality of 3D Placement on 3D Routing.

Placement	3D HPWL (m)	Routed wire length (m)	# MIVs
P1	2.43	2.85	75,750
P2	2.75	2.97	61,485
P3	2.79	3.03	77,614
P4	2.83	3.06	74,803

We built four different four-tier LDPC designs.

ACKNOWLEDGMENTS

This work was supported by the New Faculty Seed Grant funded by the Washington State University.

REFERENCES

[1] J. U. Knickerbocker, *et al.*, "Three-dimensional silicon integration," *IBM J. Res. Develop.*, vol. 52, pp. 553–569, Nov. 2008.

[2] D. H. Kim, S. Mukhopadhyay, and S. K. Lim, "TSV-aware interconnect distribution models for prediction of delay and power consumption of 3-D stacked ICs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 33, no. 9, pp. 1384–1395, Sep. 2014.

[3] D. H. Kim, K. Athikulwongse, and S. K. Lim, "Study of through-silicon-via impact on the 3D stacked IC layout," *IEEE Trans. VLSI Syst.*, vol. 21, no. 5, pp. 862–874, May2013.

[4] T. Song, S. Panth, Y.-J. Chae, and S. K. Lim, "More power reduction with 3-tier logic-on-logic 3D ICs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 35, no. 12, pp. 2056–2067, Dec. 2016.

[5] C.-H. Shen, *et al.*, "Monolithic 3D chip integrated with 500ns NVM, 3ps logic circuits and SRAM," in *Proc. IEEE Int. Electron Devices Meet.*, Dec. 2013, pp. 9.3.1–9.3.4.

[6] Y.-J. Lee and S. K. Lim, "Ultrahigh density logic designs using monolithic 3-D integration," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 32, no. 12, pp. 1892–1905, Dec. 2013.

[7] S. Bobba, *et al.*, "CELONCEL: Effective design technique for 3-D monolithic integration targeting high performance integrated circuits," in *Proc. Asia South Pacific Des. Autom. Conf.*, Jan. 2011, pp. 336–343.

[8] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Design and CAD methodologies for low power gate-level monolithic 3D ICs," in *Proc. Int. Symp. Low Power Electron. Des.*, Aug. 2014, pp. 171–176.

[9] S.-E. DavidLin, P. P. Pande, and D. H. Kim, "Optimization of dynamic power consumption in multi-tier gate-level monolithic 3D ICs," in *Proc. Int. Symp. Quality Electron. Des.*, Mar. 2016, pp. 29–34.

[10] S. Panth, K. Samadi, Y. Du, and S. K. Lim, "Power-performance study of block-level monolithic 3D-ICs considering inter-tier performance variations," in *Proc. ACM Des. Autom. Conf.*, Jun. 2014, pp. 1–6.

[11] C.-W. Lin, S.-L. Huang, K.-C. Hsu, M.-X. Lee, and Y.-W. Chang, "Multilayer obstacle-avoiding rectilinear steiner tree construction based on spanning graphs," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 27, no. 11, pp. 2007–2016, Nov. 2008.

[12] Nangate, "Nangate 45nm open cell library," (2011). [Online]. Available: http://www.nangate.com

[13] G. Karypis and V. Kumar, "hMETIS, a hypergraph partitioning package version 1.5.3," (2007). [Online]. Available: http://glaros.dte.umn.edu/gkhome/metis/hmetis/download

SHENG-EN DAVID LIN (S'16) received the BS degree in electrical engineering from Washington State University, Pullman, Washington, in 2014, where he is currently working toward the PhD degree in the Department of Electrical Engineering and Computer Science. His research interests include modeling for VLSI circuits and systems and algorithms for VLSI CAD automation with current focus on designing of monolithic 3-D ICs. He is a student member of the IEEE.

DAE HYUN KIM (S'08-M'12) received the BS degree in electrical engineering from Seoul National University, Seoul, Korea, in 2002, and the MS and PhD degrees in electrical and computer engineering from the Georgia Institute of Technology, Atlanta, Georgia, in 2007 and 2012, respectively. He is an assistant professor in the School of Electrical Engineering and Computer Science, Washington State University, Pullman, Washington. He worked on physical layout optimization at Cadence Design Systems, Inc., in San Jose, California, from 2012 to 2014. His research interests include electronic design automation and computer-aided design for VLSI, high-performance and/or low-power VLSI and computer systems, and 3-D integrated circuits and systems. He received the Defense Advanced Research Projects Agency (DARPA) Young Faculty Award (YFA) in 2016. He is a member of the IEEE.