A Legalization Algorithm for Multi-Tier Gate-Level Monolithic Three-Dimensional Integrated Circuits

Yiting Chen[1,2] and Dae Hyun Kim[1]
[1]School of Electrical Engineering and Computer Science, Washington State University, Pullman, WA, USA
[2]Schweitzer Engineering Laboratories, Pullman, WA, USA
Email: {ychen2, daehyun}@eecs.wsu.edu

***Abstract*— Design of gate-level monolithic three-dimensional integrated circuits (3-D ICs) requires 3-D placement, 3-D clock-tree synthesis, 3-D routing and monolithic inter-layer via insertion, 3-D timing and power optimization, and so on. Until now, however, various research on gate-level monolithic 3-D ICs focused on analysis of wirelength, power consumption, performance, thermal characteristics, etc. based on a design methodology using 2-D placement, uniform location scaling, z-directional partitioning, and 2-D planar legalization. However, the design of gate-level monolithic 3-D IC layouts requires more sophisticated 3-D algorithms to generate high-quality layouts. In this paper, we propose a legalization algorithm for the design of multi-tier gate-level monolithic 3-D ICs. The algorithm performs planar and z-directional legalization in an interleaved fashion to perform native 3-D legalization. We compare the proposed algorithm with a legalization algorithm being used in the literature and show that the proposed algorithm achieves shorter wirelength with almost no density constraint violation.**

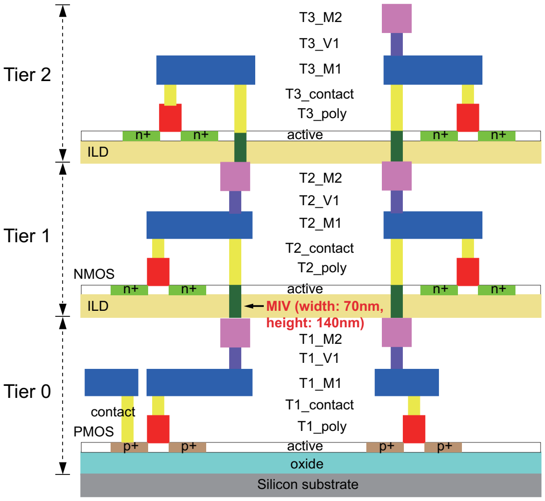

Fig. 1. Multi-tier monolithic 3-D integration.

I. Introduction

Monolithic three-dimensional integrated circuits (3-D ICs) stacks two thin device layers (tiers) and connects transistors in different tiers through monolithic inter-layer vias (MIVs) [1], [2]. Although MIVs are similar to through-silicon vias (TSVs) with respect to the shape and the functionality, the typical size of an MIV is much smaller than that of a TSV (e.g., 70nm vs. 5um in width). Due to the negligible size of MIVs, monolithic 3-D ICs provide shorter wirelength, higher performance, and lower power consumption than 2-D ICs and TSV-based 3-D ICs [3]–[5]. Multi-tier monolithic 3-D integration stacks more than two device layers as shown in Figure 1, thereby increasing the amount of benefits further [6], [7].

Design of TSV-based 3-D ICs requires 3-D placement, 3-D clock-tree synthesis, TSV insertion, 3-D routing, and so on [8]. Most of the steps, however, require careful handling of TSVs because TSVs occupy non-negligible area, so an excessive use of TSVs significantly increases the footprint area, which results in wirelength increase, performance degradation, and power consumption overhead [9], [10]. Unfortunately, minimization of the use of TSVs does not lead to a good amount of wirelength reduction, performance improvement, and power saving [11]. In addition, it is sometimes not possible to reduce the number of TSVs in the gate-level 3-D IC design [9].

Similar to the design of TSV-based 3-D ICs, the design of multi-tier gate-level monolithic 3-D ICs also requires all the design steps listed above [8]. Due to the very small size of MIVs, however, MIV insertion causes almost no area overhead. Nonetheless, the number of MIVs should also be carefully controlled because an excessive use of MIVs might lead to non-routable designs. In this paper, we propose a 3-D placement legalization algorithm to design multi-tier gate-level monolithic 3-D ICs. First, we prove that uniform-scaling-based multi-tier gate-level monolithic 3-D IC design methodology can produce high-quality layouts. Based on the proof, we propose a 3-D legalization algorithm considering the MIV count, wirelength, and instance density. We build GDSII-level multi-tier gate-level monolithic 3-D IC layouts for various benchmarks using the proposed algorithm and an hMetis-based legalization algorithm [7] and compare the quality of the algorithms.

II. Design Methodology for Multi-Tier Gate-Level Monolithic 3-D ICs

In this section, we discuss a design methodology for the design of multi-tier gate-level monolithic 3-D ICs and prove that the placement algorithm using uniform scaling can produce high-quality 3-D IC layouts.

A. Design Methodology

A design methodology proposed in the literature for the design of multi-tier gate-level monolithic 3-D ICs scales down the width and the height of each instance in a given design by a constant scaling ratio r, places the instances in a 2-D layout by a commercial tool, restores the width and the height of each instance, moves the instances across multiple tiers to spread them out along the z-direction, and performs a 2-D planar

978-1-5090-5404-6/17/$31.00 ©2017 IEEE

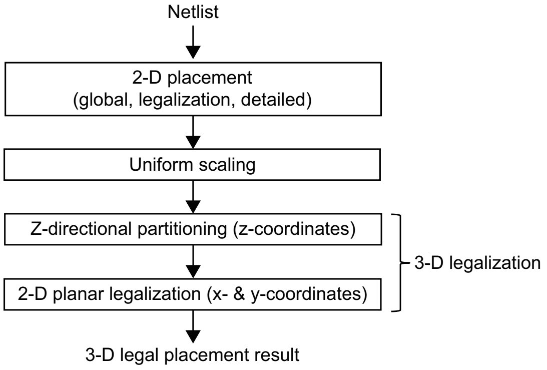
Netlist
2-D placement
(global, legalization, detailed)
Uniform scaling
Z-directional partitioning (z-coordinates)
2-D planar legalization (x- & y-coordinates)
3-D legalization
3-D legal placement result

Fig. 2. A 3-D placement methodology using uniform scaling for multi-tier gate-level monolithic 3-D IC design [7].

legalization to legalize the instances in each tier [5].[1] Another design methodology proposed in academia is based on uniform scaling [7] and shown in Figure 2. This so-called *uniform-scaling-based placement* places instances in a 2-D layout by a commercial tool, scales down the locations of the instances by a constant scaling ratio r, moves the instances across multiple tiers to spread them out along the z-direction, and performs a 2-D planar legalization similar to the design methodology in [5]. The second methodology is simpler than the first and guarantees that final 3-D layouts are high-quality, which will be proved in Section III.

B. Z-Directional Partitioning and 2-D Planar Legalization

The two algorithms described above move instances across multiple tiers to spread them out along the z-direction after the instance locations are finalized in 2-D. We call this step *z-directional partitioning* in this paper. The objective function they use in the z-directional partitioning step is the total cut size, which is somewhat proportional to the total MIV count. In addition, they apply local density constraints (i.e., the density of each bin is less than a pre-determined value) so that the instances are evenly spread out across the tiers. If the density constraints are globally applied to the entire layout (i.e., the density of each tier is less than a pre-determined value), it could happen in the worst case that all the instances in the bottom half of Tier 0 in the initial layout are moved to Tier 1 when it is designed in two tiers. In this case, the density of each tier satisfies the density constraint, but the 2-D planar legalization after the z-directional partitioning needs to distribute the instances placed in the bottom half of Tier 1 to the top half in the same tier and the instances placed in the top half of Tier 0 to the bottom half in the same tier. Then, the total wirelength will significantly increase. To avoid this kind of situations, the density constraint is applied locally so that the instances can be distributed well over the entire layout and the tiers.

The 2-D planar legalization used by the two algorithms after z-directional partitioning legalizes instance locations without considering 3-D nets. In other words, the instances in each tier are legalized (snapping and overlap removal) without the information of the locations of the instances in other tiers. In this case, the wirelength of 2-D nets as well as 3-D nets might not be fully optimized.

C. Motivation of This Work

The design methodology using uniform scaling has several problems. First, it optimizes the total cutsize, but not the displacement of each instance because the z-directional partitioning does not consider the overlaps among the instances. Ignoring the displacement of each instance during the z-directional partitioning could lead to a non-negligible wirelength increase for some nets. Second, the 2-D planar legalization ignores the connectivity of 3-D nets, so it might increase the wirelength of 2-D and 3-D nets. Nonetheless, the uniform-scaling-based 3-D placement algorithm has a big advantage that the wirelength of a final 3-D layout obtained by the algorithm will be near-optimal if the wirelength of the initial 2-D layout is optimal as will be shown in the next section. Thus, we propose a new 3-D legalization algorithm considering the total cutsize and instance overlaps at the same time.

III. Optimality of the Uniform-Scaling-Based Placement

In this section, we analyze the optimality of the wirelength reduction (i.e., wirelength minimization) of the uniform-scaling-based 3-D placement algorithm.

A. Uniform-Scaling-Based 3-D Placement

When a legal 2-D placement result L_{2D} is obtained in Figure 2, the location of the bottom-left corner, the width, and the height of each instance c_i in the layout are represented by (x_i, y_i), w_i, and h_i, respectively. Notice that all the instances have the same height. For L_{2D}, we obtain a non-legal 3-D placement result $L_{\mathrm{3D,NL}}$ by downscaling the location of each instance by a constant scaling ratio r. Thus, the location of the bottom-left corner of instance c_i in $L_{\mathrm{3D,NL}}$ after uniform scaling becomes $(r \cdot x_i, r \cdot y_i, z_i = 0)$ in which we explicitly show the z-coordinate. The algorithm sets r to $1/\sqrt{N_{\mathrm{T}}}$ where N_{T} is the number of tiers. z_i of the bottommost tier is 0 and that of the topmost tier is $N_{\mathrm{T}} - 1$.

Some instances in Tier 0 in the non-legal 3-D placement result $L_{\mathrm{3D,NL}}$ will overlap, so the uniform scaling is followed by z-directional partitioning, which moves some of the instances to the other tiers. This process sets the z-coordinates of the instances moved to Tier k to k. Then, 2-D planar legalization is performed to snap the instances into standard cell rows and completely remove the instance overlaps in each tier.

B. Optimality: Ideal Case

When we obtain a legal 3-D placement result $L_{\mathrm{3D,L}}$ from $L_{\mathrm{3D,NL}}$ by planar legalization, a fundamental question about the wirelength optimality (i.e., whether the wirelength is minimal) of the 3-D placement algorithm arises. In this

[1] In “2-D planar legalization”, “2-D” means that it considers only 2-D connections (i.e., ignores 3-D connections) and “planar legalization” means that instances are moved in each tier, but not across tiers (i.e., the x- and y-coordinates of the instances are changed, but the z-coordinates are not changed.)

section, we show the wirelength optimality in ideal cases by which we mean that we perform z-directional partitioning and obtain $L_{\mathrm{3D,NL}}$, but we do not need planar legalization after the z-directional partitioning. In other words, the location of instance c_i in $L_{\mathrm{3D,L}}$ is just $(r \cdot x_i, r \cdot y_i, z_i)$.[2]

The question can be rephrased as follows: "Suppose a given initial legal 2-D placement result L_{2D} has the shortest total wirelength (i.e., L_{2D} is optimal in terms of wirelength). Then, will the 3-D placement result $L_{\mathrm{3D,NL}}$ obtained by uniformly scaling and z-directional partitioning also have the shortest total wirelength if the instances need no planar legalization?" and the answer is yes. In the proof shown below, we use the half-perimeter wirelength (HPWL) to approximate the length of each net.

Proof: Suppose a given 2-D placement result L_{2D} is optimal, i.e., it has the shortest wirelength. The total wirelength of L_{2D} can be represented as follows:

$$\mathrm{HPWL}(L_{\mathrm{2D}}) = \sum_{\forall n \in N(L_{\mathrm{2D}})} \mathrm{HPWL}(n) \tag{1}$$

where $N(L_{\mathrm{2D}})$ is the set of all nets of L_{2D} and $\mathrm{HPWL}(n)$ is the HPWL of net n. Similarly, we obtain the total wirelength of $L_{\mathrm{3D,NL}}$ as follows:

$$\mathrm{HPWL}(L_{\mathrm{3D,NL}}) = \sum_{\forall n \in N(L_{\mathrm{3D,NL}})} \mathrm{HPWL}_{\mathrm{2D}}(n) \tag{2}$$

where $\mathrm{HPWL}_{\mathrm{2D}}(n)$ is the 2-D (planar) HPWL of net n. In other words, we project all the instances connected to net n onto a 2-D plane and obtain the HPWL of the net in the plane. The reason that we do not add the z-directional length (the sum of the MIV length and the via stack length) to the HPWL is because the z-directional wirelength of a 3-D net is in general much shorter than the planar wirelength of the net.

Uniform scaling preserves the relative locations of the instances, so the following equation holds:

$$\mathrm{HPWL}(L_{\mathrm{2D}}) = \frac{1}{r} \cdot \mathrm{HPWL}(L_{\mathrm{3D,NL}}) \tag{3}$$

If $L_{\mathrm{3D,NL}}$ is not optimal, there exists a layout $L'_{\mathrm{3D,NL}}$ such that $\mathrm{HPWL}(L'_{\mathrm{3D,NL}})$ is less than $\mathrm{HPWL}(L_{\mathrm{3D,NL}})$. Then, we can obtain a 2-D placement result L'_{2D} by upscaling the locations of the instances in $L'_{\mathrm{3D,NL}}$ by $1/r$ and setting the z-coordinates of all the instances to 0. Since the upscaling also preserves the relative locations of the instances, the following equation holds:

$$\mathrm{HPWL}(L'_{\mathrm{2D}}) = \frac{1}{r} \cdot \mathrm{HPWL}(L'_{\mathrm{3D,NL}}). \tag{4}$$

Since $L'_{\mathrm{3D,NL}}$ has a shorter wirelength than $L_{\mathrm{3D,NL}}$, the following inequality holds:

$$\mathrm{HPWL}(L'_{\mathrm{2D}}) = \frac{1}{r} \cdot \mathrm{HPWL}(L'_{\mathrm{3D,NL}}) < \frac{1}{r} \cdot \mathrm{HPWL}(L_{\mathrm{3D,NL}}) = \mathrm{HPWL}(L_{\mathrm{2D}}), \tag{5}$$

which is a contradiction because we assumed that L_{2D} is optimal. Thus, if L_{2D} has the shortest wirelength, $L_{\mathrm{3D,NL}}$ obtained by the uniform scaling of the instance locations also has the shortest wirelength in 3-D.

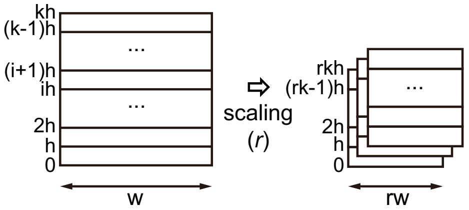

Fig. 3. Standard cell rows in gate-evel 2-D (left) and 3-D (right) layouts.

C. Optimality: Non-Ideal Case

In this section, we answer the question "How optimal is $L_{\mathrm{3D,L}}$?" assuming non-ideal cases. Here, we mean by "non-ideal cases" that we perform planar legalization after z-directional partitioning for snapping and overlap removal. In this case, if the location of instance c_i in L_{2D} is (x_i, y_i), its location in $L_{\mathrm{3D,L}}$ obtained by uniform scaling and planar legalization is $(r \cdot x_i + \alpha_i, r \cdot y_i + \beta_i, z_i)$ where α_i and β_i are real numbers introduced to represent the offset caused by the planar legalization.

If the offset of an instance is very small compared to the width and the height of the instance, i.e., $\alpha_i \ll w_i$ and $\beta_i \ll h_i$, then the location of the instance in $L_{\mathrm{3D,L}}$ can be represented by $(r \cdot x_i, r \cdot y_i, z_i)$, which is equal to its location in the ideal case, so $L_{\mathrm{3D,L}}$ is optimal. If α_i and β_i are not negligible, however, $\mathrm{HPWL}(L_{\mathrm{3D,L}})$ might be greater than the total wirelength of an optimal 3-D placement result and we quantify the deviation in this section.

When the planar legalization is performed, we still assume that we can preserve the relative locations of the instances. Mathematically, preservation of the relative locations can be represented as follows. Suppose the locations of two instances, c_1 and c_2, placed in L_{2D} are (x_1, y_1) and (x_2, y_2), respectively, and their locations in $L_{\mathrm{3D,L}}$ after uniform scaling and legalization are $(r \cdot x_1 + \alpha_1, r \cdot y_1 + \beta_1, z_1)$ and $(r \cdot x_2 + \alpha_2, r \cdot y_2 + \beta_2, z_2)$, respectively. Then, the preservation condition guarantees (without loss of generality) that 1) if $x_1 \leq x_2$, then $r \cdot x_1 + \alpha_1 \leq r \cdot x_2 + \alpha_2$ and 2) if $y_1 \leq y_2$, then $r \cdot y_1 + \beta_1 \leq r \cdot y_2 + \beta_2$.[3]

1) Y-Coordinates: In Fig. 3, a 2-D layout whose width is w and height is $k \cdot h$ is given. h is the height of a standard cell row, so there are k rows in the layout. After uniform scaling with a scaling ratio $r = 1/\sqrt{N_{\mathrm{T}}}$, the width and the height of the 3-D layout become $r \cdot w$ and $r \cdot k \cdot h$, respectively, and each row in the 3-D layout has $N_{\mathrm{T}} - 1$ tiers on top of it (total N_{T} tiers).

The area of each row in the 2-D layout is $w \cdot h$ and the sum of the areas of the rows in the 3-D layout is $r \cdot w \cdot h \cdot N_{\mathrm{T}}$ = $w \cdot h \cdot \sqrt{N_{\mathrm{T}}}$. When the y-coordinate of an instance in the 2-D layout is $i \cdot h (0 \leq i \leq k-1)$, its location in the 3-D layout before legalization becomes $r \cdot i \cdot h$. We show an

[2] In reality, planar legalization is mandatory because the locations of some of the instances cannot be aligned with the standard cell rows and columns after uniform scaling even if there is no overlap between any two instances.

[3] This does not hold in some cases, especially when the density of a small area is very high and there are very wide instances in the area. However, we assume that whitespace is well distributed over the entire layout area and the width of the widest instance is much smaller than the width of the layout. In this case, we can practically assume that the relative locations of the instances can be preserved.

TABLE I

A SCALING EXAMPLE. y_{2D} IS THE Y-COORDINATE OF AN INSTANCE IN A 2-D LAYOUT AND y_{3D} IS THE Y-COORDINATE OF THE INSTANCE AFTER SCALING AND SNAPPING INTO ONE OF THE STANDARD CELL ROWS THE INSTANCE IS OVERLAPPING WITH. $N_{\mathrm{T}} = 8$. $r = 1/\sqrt{N_{\mathrm{T}}}$.

y_{2D}	$r \cdot y_{\mathrm{2D}}$	y_{3D}	y_{2D}	$r \cdot y_{\mathrm{2D}}$	y_{3D}
0	0	0	5	1.76777	1, 2
1	0.35355	0	6	2.12132	2
2	0.70711	0, 1	7	2.47487	2
3	1.06066	1	8	2.82843	2, 3
4	1.41421	1	9	3.18198	3

outline of a proof that we can partition the instance along the z-direction and snap it to the row it is overlapped with, i.e., its y-coordinate after z-directional partitioning can be $\lfloor r \cdot i \rfloor \cdot h$ or $(\lfloor r \cdot i \rfloor + 1) \cdot h$.

Outline of a Proof: The area of each row in the 3-D layout is $w \cdot h \cdot \sqrt{N_{\mathrm{T}}}$. Thus, the first $\lfloor \sqrt{N_{\mathrm{T}}} \rfloor$ rows in the 2-D layout can be completely placed into the first row in the 3-D layout and some of the instance in the $(\lfloor \sqrt{N_{\mathrm{T}}} \rfloor + 1)$-th row can be placed into the first row in $L_{\mathrm{3D,L}}$ and the rest of the instances in the row can be placed into the second row in the 3-D layout. Similarly, if the y-coordinate of an instance in the 2-D layout is $i \cdot h$, its y-coordinate before z-directional partitioning in the 3-D layout is $r \cdot i \cdot h = i \cdot h/\sqrt{N_{\mathrm{T}}}$. Since the area of each row in the 3-D layout is $w \cdot h \cdot \sqrt{N_{\mathrm{T}}}$, the y-coordinate of the instance in the 3-D layout can be legalized into either $\lfloor r \cdot i \rfloor \cdot h$ or $(\lfloor r \cdot i \rfloor + 1) \cdot h$.

An example is shown in Table I in which N_{T} is 8. In the table, we verify that the y-coordinate of an instance after uniform scaling and snapping into one of the rows the instance is overlapping with is either $\lfloor r \cdot y_{\mathrm{2D}} \rfloor \cdot h$ or $(\lfloor r \cdot y_{\mathrm{2D}} \rfloor + 1) \cdot h$. Thus, when the coordinate of an instance in the 2-D layout is (x_i, y_i) and its y-coordinate after uniform scaling, z-directional partitioning, and y-coordinate snapping is $r \cdot y_i + \beta_i$, β_i is less than h that is the height of a standard cell row.

Notice that the proof shown above does not mean that the y-coordinate of instance c_i located at $(x_i, y_i, 0)$ in the initial 2-D layout should be $\lfloor r \cdot i \rfloor \cdot h$ or $(\lfloor r \cdot i \rfloor + 1) \cdot h$ after uniform scaling and snapping. Rather, it means that the utilization of each standard cell row in the 3-D layout will be less than 1.0 after uniform scaling and snapping if the instances are properly snapped into one of its adjacent rows.

2) X-Coordinates: The offsets of the x-coordinates of the instances can be similarly derived as those of the y-coordinates. However, the width of each standard cell varies, whereas the height of them is a constant. In this paper, however, we assume without a proof that the maximum displacement of each instance after uniform scaling and planar legalization is $w_{\mathrm{m}}/2$ where w_{m} is the width of the maximum-width standard cell. Thus, the range of α_i is $[0, h/2]$ and that of β_i is $[0, w_{\mathrm{m}}/2]$. Applying this to each instance, the upper bound of $\mathrm{HPWL}_{\mathrm{3D}}(n)$ of net n in $L_{\mathrm{3D,L}}$ will be $r \cdot \mathrm{HPWL}_{\mathrm{2D}}(n) + h + w_{\mathrm{m}}$.

IV. LEGALIZATION ALGORITHM

In this section, we propose a legalization algorithm for multi-tier gate-level monolithic 3-D IC design. Figure 4 shows the algorithm. For a given netlist, we perform 2-D placement using a commercial tool and uniform scaling with a scaling ratio $1/\sqrt{N_{\mathrm{T}}}$. Then, we perform snapping, binning, z-directional partitioning, row adjustment, and single-row legalization as explained in detail below.

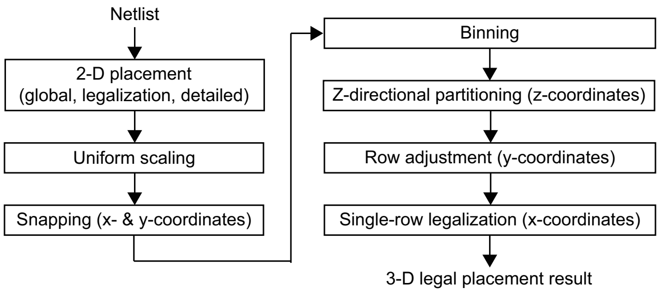
Netlist
2-D placement
(global, legalization, detailed)
Uniform scaling
Snapping (x- & y-coordinates)
Binning
Z-directional partitioning (z-coordinates)
Row adjustment (y-coordinates)
Single-row legalization (x-coordinates)
3-D legal placement result

Fig. 4. Our legalization algorithm.

A. Snapping

Uniform scaling changes the location of instance c_i from $(x_i, y_i, 0)$ to $(r \cdot x_i, r \cdot y_i, 0)$. Since the initial location $(x_i, y_i, 0)$ is a legal location, it can be represented as $(p_i \cdot w_{\mathrm{r}}, q_i \cdot h_{\mathrm{r}}, 0)$ where p_i and q_i are integers and w_{r} and h_{r} are the x- and y-directional step size, respectively. In other words, w_{r} is the width of a standard cell column and h_{r} is the height of a standard cell row. Then, the instance location after uniform scaling is $(r \cdot p_i \cdot w_{\mathrm{r}}, r \cdot q_i \cdot h_{\mathrm{r}}, 0)$. Then, we perform row and column snapping to snap the instances into the standard cell rows and columns. The row snapping snaps instance c_i into the closer row between $\lfloor r \cdot q_i \rfloor \cdot h_{\mathrm{r}}$ and $(\lfloor r \cdot q_i \rfloor + 1) \cdot h_{\mathrm{r}}$. If the distance from the instance to $\lfloor r \cdot q_i \rfloor \cdot h_{\mathrm{r}}$ is equal to the distance to $(\lfloor r \cdot q_i \rfloor + 1) \cdot h_{\mathrm{r}}$, we choose the former row. This might lead to $> 100\%$ utilization in some rows, but we resolve this issue in the row adjustment step.

B. Binning

Suppose the width and the height of the target 3-D layout are w_{L} and h_{L}, respectively. Then, we gridize the layout into bins where the size of each bin $w_{\mathrm{b}} \times w_{\mathrm{b}}$ is pre-determined and assign the instances to the bins based on the coordinates of the bottom-left corners of the instances. The identification triplet of each bin is (b_l, b_m, b_n) where b_l, b_m, and b_n are integers. Initially, the utilization of bin (b_l, b_m, b_n) is zero if $b_n > 0$ and approximately $u_0 \cdot N_{\mathrm{T}}$ where u_0 is the average utilization of the original 2-D layout. Depending on the distribution of the instances in the 2-D layout, however, the initial utilization of bin $(b_l, b_m, 0)$ might be greater than $u_0 \cdot N_{\mathrm{T}}$. Thus, the maximum utilization of bin (b_l, b_m, b_n) is set to $U_0(l, m, 0)/N_{\mathrm{T}} + \delta_{\mathrm{u}}$ where $U_0(l, m, n)$ is the initial utilization of bin (b_l, b_m, b_0) and δ_{u} is a margin to allow moving wide instances.

C. Z-Directional Partitioning

Suppose the location of instance c_i after snapping is $(x_i, y_i, 0)$. Then, z-directional partitioning moves the cell to (x_i, y_i, z_i) where z_i is an integer and varies from 0 to $N_{\mathrm{T}} - 1$. The goal of the z-directional partitioning is to satisfy the

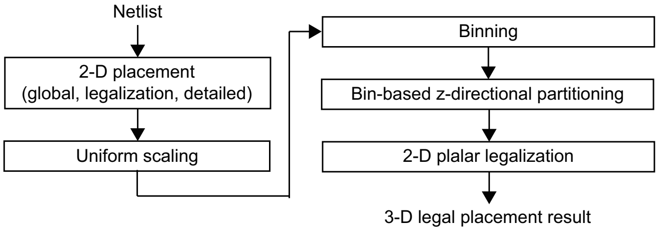
Netlist
2-D placement
(global, legalization, detailed)
Uniform scaling
Binning
Bin-based z-directional partitioning
2-D plalar legalization
3-D legal placement result

Fig. 5. The 3-D legalization algorithm in [7].

density constraint of each bin while minimizing the total cutsize and the overlaps among the instances for the minimization of the number of MIVs and the instance displacement, respectively. We perform the Fiduccia-Mattheyses (FM) partitioning algorithm for the z-directional partitioning. The objective (cost) function is as follows:

$$F = w_1 \cdot \sum_{\forall n \in N} cs(n) + w_2 \cdot \sum_{\forall i \in C} \sum_{\forall j \in C, i \neq j} ov(c_i, c_j) \quad (6)$$

where N is the set of nets, $cs(n)$ is the cutsize of Net n, C is the set of instances, $ov(c_i, c_j)$ is the overlap area between instances c_i and c_j, and w_1 and w_2 are weighting factors. The cutsize of Net n is computed by the difference between the topmost tier and the bottommost tier that the instances connected to the net belong to.

When we compute the gain of moving an instance from its current tier to another tier, we also check whether the move will violate the density constraint of the target tier. If it violates, we do not accept the move. In addition, we also check whether the move will violate a pre-determined maximum utilization of each standard cell row that the instance belongs to. If it violates the constraint, we do not accept the move.

D. Row Adjustment and Single-Row Legalization

The final step of our legalization algorithm is to legalize instance locations in each standard cell row (single-row legalization) preserving the relative instance locations. If the utilization of a standard cell row is too high, however, the single-row legalization cannot be applied. Thus, we slightly adjust the y-coordinates of some instances before we apply the single-row legalization. The row adjustment algorithm is as follows. First, we choose a highest-utilization row g. If the utilization of g is less than a pre-determined value (e.g., 0.8), we stop row adjustment and proceed to the single-row legalization. Otherwise, we randomly choose an instance in the row and move it to one of the rows adjacent to g. Since a standard cell row is generally adjacent to two rows (above or below), we choose the row that has lower utilization than the other between the two. Once all the rows in the 3-D layout satisfies the row-density constraint, we perform single-row legalization using the clumping algorithm [12].

V. Simulation Results

In this section, we present our simulation results and detailed analysis.

A. Simulation Settings

We use the Nangate 45nm standard cell library for our simulation. Synopsys Design Compiler is used for netlist synthesis and Cadence Innovus is used for 2-D placement. We also compare our algorithm with the 3-D placement algorithm presented in [7]. Briefly speaking, the algorithm also performs uniform scaling, binning, bin-based z-directional partitioning, and 2-D planar legalization as shown in Figure 5. The bin-based z-directional partitioning performs k-way partitioning to partition the instances in bin $(b_l, b_m, 0)$ into $(b_l, b_m, 0)$, ..., $(b_l, b_m, k-1)$ where k is N_{T}. It sequentially performs the partitioning starting from bin $(0, 0, 0)$. When the instances in $(b_l, b_m, 0)$ are partitioned into the N_{T} bins (tiers), it also takes the instances that have already been partitioned into account. In other words, if an instance c_i in $(b_l, b_m, 0)$ is connected to an instance c_j in another bin and the tier of c_j has already been determined, it considers c_j as a fixed pin when it partitions the bin c_i belongs to. We call this algorithm *2-DL* and the proposed algorithm *3-DL* for simplicity in this section.

B. Wirelength and Cutsize

Table II shows HPWL values of 2-DL and 3-DL for four benchmark circuits and seven different designs (two- to eight-tier designs) for each benchmark circuit. For all the benchmarks, 3-DL achieves shorter HPWL than 2-DL at the cost of increased cutsize in most cases.

For Ckt1, the wirelength of 3-DL is always shorter than that of 2-DL for all the two- to eight-tier layouts. In addition, for two- and three-tier layouts, 3-DL achieves smaller cutsize than 2-DL. Moreover, 3-DL achieves almost no density violation, whereas 8 to 37 bins violate the density constraint in the result of 2-DL. If we fix the density violation problem in the layout generated by 2-DL, either the HPWL or the cutsize or both of them will increase. The rightmost column in the table shows the HPWL ratios between the layouts build by 3-DL and the 2-D layouts. As the numbers and $1/\sqrt{N_{\mathrm{T}}}$ in the 10-th column in the table show, our results are very close to the ideal wirelength reduction ratio values.

For Ckt2, the two algorithms show similar wirelength, but 3-DL achieves 19% to 23% less cutsize with almost no density violation. However, the difference between the ideal wirelength ratio $(1/\sqrt{N_{\mathrm{T}}})$ and the actual wirelength reduction (2-D HPWL / Ours) is greater than that for Ckt1.

3-DL has much shorter wirelength than 2-DL for several Ckt3 and Ckt4 designs such as the three-tier Ckt3 and six- and seven-tier Ckt4 designs. It is because 3-DL is stable, i.e., consistently reduces the wirelength efficiently, but 2-DL does not consistently produce high-quality layouts. Thus, 3-DL produces 14% to 28% shorter wirelength results in some cases. However, 2-DL achieves smaller cutsize than 3-DL for large designs (Ckt3 and Ckt4). Assuming each MIV requires 0.28um × 0.28um area for routing, the MIVs occupy less than 2.5% area in the worst case (the eight-tier design of Ckt3), which is acceptable. In addition, many bins in the 2-DL designs violate the density constraint, while only a few bins violate the density constraint in the 3-DL designs. The reason that 3-DL also generates layouts having density violations is

TABLE II

Comparison of 3-D legalization algorithms. # DVs is the number of bins violating the density constraint. 2-D HPWL of each benchmark is the HPWL of the 2-D layout of the design.

Benchmark	2-D HPWL (mm)	# tiers	[7]			Ours			$\frac{1}{\sqrt{N_\mathrm{T}}}$	Ours / 2-D HPWL
			HPWL (mm)	Cutsize	# DVs	HPWL (mm)	Cutsize	# DVs		
Ckt1	129	2	99 (1.000)	3,317 (1.000)	8	98 (**0.990**)	2,876 (**0.867**)	0	0.707	0.715
		3	86 (1.000)	5,937 (1.000)	21	86 (**1.000**)	5,715 (**0.963**)	1	0.577	0.588
		4	79 (1.000)	8,457 (1.000)	26	76 (**0.962**)	8,653 (**1.023**)	1	0.500	0.512
		5	75 (1.000)	10,554 (1.000)	32	71 (**0.947**)	11,148 (**1.056**)	1	0.447	0.461
		6	71 (1.000)	12,140 (1.000)	24	69 (**0.972**)	13,939 (**1.148**)	0	0.408	0.424
		7	69 (1.000)	14,835 (1.000)	38	63 (**0.913**)	16,519 (**1.114**)	0	0.378	0.394
		8	67 (1.000)	16,209 (1.000)	37	65 (**0.970**)	18,488 (**1.141**)	1	0.354	0.371
Ckt2	634	2	460 (1.000)	9,993 (1.000)	78	466 (**1.013**)	7,686 (**0.769**)	0	0.707	0.742
		3	383 (1.000)	19,429 (1.000)	130	384 (**1.003**)	15,176 (**0.781**)	0	0.577	0.637
		4	339 (1.000)	27,370 (1.000)	153	335 (**0.988**)	22,165 (**0.810**)	0	0.500	0.556
		5	310 (1.000)	35,356 (1.000)	197	307 (**0.990**)	27,320 (**0.773**)	7	0.447	0.545
		6	288 (1.000)	43,979 (1.000)	205	283 (**0.983**)	33,967 (**0.772**)	3	0.408	0.538
		7	273 (1.000)	52,234 (1.000)	217	265 (**0.971**)	40,035 (**0.766**)	6	0.378	0.502
		8	261 (1.000)	59,313 (1.000)	222	252 (**0.966**)	47,619 (**0.803**)	4	0.354	0.476
Ckt3	978	2	740 (1.000)	17,456 (1.000)	201	725 (**0.980**)	23,716 (**1.359**)	1	0.707	0.764
		3	721 (1.000)	27,052 (1.000)	522	623 (**0.864**)	40,618 (**1.501**)	2	0.577	0.668
		4	592 (1.000)	31,743 (1.000)	662	543 (**0.917**)	53,364 (**1.681**)	1	0.500	0.592
		5	539 (1.000)	31,317 (1.000)	454	533 (**0.988**)	60,273 (**1.925**)	1	0.447	0.549
		6	547 (1.000)	35,475 (1.000)	448	526 (**0.962**)	72,922 (**2.056**)	4	0.408	0.537
		7	500 (1.000)	35,824 (1.000)	396	491 (**0.983**)	81,851 (**2.285**)	4	0.378	0.490
		8	508 (1.000)	42,749 (1.000)	453	466 (**0.917**)	88,834 (**0.917**)	7	0.354	0.505
Ckt4	4,304	2	3,068 (1.000)	11,565 (1.000)	63	3,078 (**0.997**)	13,634 (**1.179**)	1	0.707	0.735
		3	2,599 (1.000)	16,526 (1.000)	196	2,530 (**0.989**)	23,095 (**1.397**)	2	0.577	0.606
		4	2,264 (1.000)	24,391 (1.000)	248	2,203 (**0.973**)	32,484 (**1.332**)	9	0.500	0.530
		5	2,179 (1.000)	29,381 (1.000)	314	1,985 (**0.911**)	41,500 (**1.412**)	16	0.447	0.484
		6	2,203 (1.000)	34,199 (1.000)	302	1,823 (**0.828**)	51,735 (**1.513**)	10	0.408	0.446
		7	2,325 (1.000)	43,725 (1.000)	341	1,695 (**0.729**)	60,647 (**1.387**)	14	0.378	0.417
		8	1,606 (1.000)	49,261 (1.000)	187	1,595 (**0.993**)	69,484 (**1.411**)	18	0.354	0.397

because the row adjustment and the single-row legalization steps in Figure 4 do not strictly check the density constraint in each bin. As a result, some bins in the layouts generated by 3-DL violate the density constraint, but the violation amount[4] is small (only 1% to 2%) and the density violations can be easily fixed by a post-process.

In summary, the proposed algorithm (3-DL) produces high-quality multi-tier gate-level monolithic 3-D IC layouts with acceptable cutsize and almost no density violation, whereas the legalization algorithm in [7] (2-DL) is less reliable and generates layouts having longer wirelength and many density violations.

VI. Conclusion

In this paper, we proposed a legalization algorithm for the design of multi-tier gate-level monolithic 3-D ICs. The algorithm performs planar and z-directional legalization in an interleaved fashion while minimizing the total wirelength and the displacement of the instances. The simulation results show that the proposed algorithm achieves shorter wirelength with almost no density constraint violation compared to the legalization algorithm used in [7].

References

[1] C.-H. Shen, J.-M. Shieh, T.-T. Wu, W.-H. Huang, C.-C. Yang, *et al.*, “Monolithic 3D Chip Integrated with 500ps NVM, 3ps Logic Circuits and SRAM,” in *Proc. IEEE Int. Electron Devices Meeting*, Dec. 2013, pp. 9.3.1–9.3.4.

[2] MonolithicI3D, http://monolithic3d.com.

[3] Y.-J. Lee and S. K. Lim, “Ultrahigh Density Logic Designs Using Monolithic 3-D Integration,” in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 32, no. 12, Dec. 2013, pp. 1892–1905.

[4] 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 Design Automation Conf.*, June 2014, pp. 1–6.

[5] 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. on Low Power Electronics and Design*, Aug. 2014, pp. 171–176.

[6] S. Bobba, A. Chakraborty, O. Thomas, P. Batude, T. Ernst, *et al.*, “CELONCEL: Effective Design Technique for 3-D Monolithic Integration Targeting High Performance Integrated Circuits,” in *Proc. Asia and South Pacific Design Automation Conf.*, Jan. 2011, pp. 336–343.

[7] S.-E. D. Lin, P. P. Pande, and D. H. Kim, “Optimization of Dynamic Power Consumption in Multi-Tier Gate-Level Monolithic 3D ICs,” in *Proc. Int. Symp. on Quality Electronic Design*, Mar. 2016, pp. 29–34.

[8] D. H. Kim and S. K. Lim, “Physical Design and CAD Tools for 3-D Integrated Circuits: Challenges and Opportunities,” in *IEEE Design and Test*, vol. 32, no. 4, Aug. 2015, pp. 8–22.

[9] D. H. Kim, K. Athikulwongse, and S. K. Lim, “Study of Through-Silicon-Via Impact on the 3D Stacked IC Layout,” in *IEEE Trans. on VLSI Systems*, vol. 21, no. 5, May 2013, pp. 862–874.

[10] 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,” in *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 33, no. 9, Sept. 2014, pp. 1384–1395.

[11] D. H. Kim, S. Mukhopadhyay, and S. K. Lim, “TSV-aware Interconect Length and Power Prediction for 3D Stacked ICs,” in *Proc. IEEE Int. Interconnect Technology Conference*, June 2009, pp. 26–28.

[12] A. B. Kahng, P. Tucker, and A. Zelikovsky, “Optimization of Linear Placements for Wirelength Minimization with Free Sites,” in *Proc. Asia and South Pacific Design Automation Conf.*, Jan. 1999, pp. 241–244.

[4] A violation amount for each bin is defined as $(A - M) * 100/B$ where A is the area of the bin, M is the density constraint (max. utilization), and B is the bin area.