General Floorplanning Methodology for 3D ICs with An Arbitrary Bonding Style

Jai-Ming Lin and Chien-Yu Huang
Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan
Email: jmlin@ee.ncku.edu.tw and joy7160076@gmail.com

***Abstract*—This paper proposes a general floorplanning methodology which can be applied to 3D ICs with an arbitrary bonding style. Some researches have shown that a 3D IC with the hybrid bonding style, which includes face-to-back and face-to-face, may obtain better results than that simply using the face-to-back bonding style. We respectively present an approach to assign modules to tiers for each kind of bonding style. Further, a new utilization function, called cosine-shaped function, is proposed to estimate utilizations of bins required by the analytical-based approach. Our experimental results show the cosine-shaped function can obtain a little better result than the bell-shaped function on IBM benchmarks for 2D floorplanning. We also show that the proposed 3D floorplanning methodology consumes less TSVs and induces shorter wirelength compared to previous work in the hybrid bonding style.**

I. Introduction

As great strides have been made in manufacturing techniques, 3D integration is considered as a promising solution in producing ICs for the next generation. Since locations of modules have great impact on wirelength and power consumption, floorplanning still plays a critical role in 3D ICs.

The bonding style [1] in 3D ICs can be divided into three categories, which include face-to-face (F2F), face-to-back (F2B) and back-to-back (B2B), respectively. Tiers with the F2B or the B2B bonding styles are connected by through-silicon vias (TSVs). TSVs will occupy large placement areas and cause longer signal delays. In stead of using TSVs, F2F style uses vias to connect modules in different tiers. The area and parasitic of a via is much small than a TSV. Hence, F2F requires less placement area and its signal delay may be reduced. F2B is a most commonly used bonding style in 3D ICs because it can continuously stack multiple tiers without wasting more TSVs like the B2B bonding style, which will result in higher manufacturing costs. Hence, B2B bonding style is rarely used. Although the F2F bonding style has superior properties than other styles, it is constrained in the top two tiers of a 3D IC. Otherwise, it will result in the B2B bonding style in other tiers.

Early works use the simulated annealing (SA) algorithm with a representation to handle 3D floorplanning [6], [12], [13]. Li *et al.* [8] use the generalized slicing tree to legalize modules and TSVs in each tier of a 3D IC, called Co-place. Lin and Yang *et al.* [10] propose an analytical based approach to handle 3D floorplanning. Recently, researches [4], [11] have compared power consumption and wirelength required by 3D ICs in the classical F2B bonding style with those in 3D ICs in the hybrid bonding style, which comprises the F2B and F2F simultaneously. The experimental results show that the power consumption and the wirelength of a design implemented in a 3D IC with the hybrid style can be respectively reduced by 1.5% and 2.2% compared to that in the F2B bonding style. Fig. 1 shows 3D ICs with the classic F2B bonding style and the hybrid bonding style. Note that the tiers in the top two tiers are connected by the F2F bonding style while the tiers in the lower two tiers are connected by the F2B bonding style in Fig. 1(b). Because the hybrid bonding style in Fig. 1(b) requires less signal delivery TSVs than that in Fig. 1(a), it will have better performance and lower cost.

This work was partially supported by the National Science Council of Taiwan ROC under Grant No. MOST 106-2221-E-006-236-MY2.

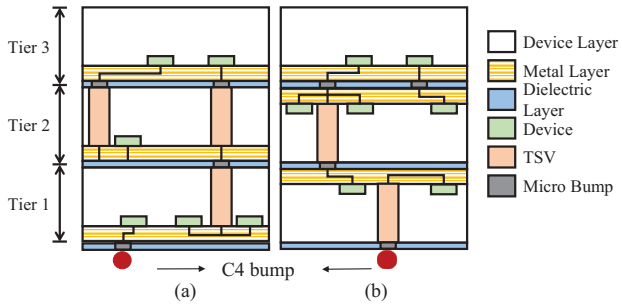

Fig. 1. A 3D IC with (a) classical F2B bonding style, and (b) hybrid bonding style.

This paper proposes a methodology which can be applied to handle 3D floorplanning with any bonding style. Since modules allocated to tiers with different bonding styles have different area requirements and could induce various signal delays, it is necessary to have different approaches to gain best benefits from respective architecture. Hence, this paper respectively proposes a partition-based and an analytical-based approaches to allocate modules to tiers for the two kind bonding styles.

Utilization function is one of the most important components in the analytical-based method, which requires it to calculate the module area in each bin after dividing a region into regular bins. Fig. 2(a) shows an original utilization function, where w_v denotes the width of a module and w_b denotes the width of a bin. The x-axis denotes the distance between a module and a bin, and the y-axis denotes the utilization of modules with respect to the bin. The utilization function is neither smooth nor differentiable, which is not applicable to the analytical-based approach. Hence, Kahng and Wang [5] propose to apply the bell-shaped function to calculate utilization of a bin (see Fig. 2(b)). Although the bell-shaped function is smooth and differentiable, there still exists an error between the original utilization function. See the curve in Fig. 2(b) and the curve in Fig. 2(a). Hence, Hsu *et al.* [3] propose to use a sigmoid function as an utilization function (see Fig. 2(c) for example). Although the curve of the sigmoid function looks much similar to the original overlap function, the curve close to the center is too flat which makes it fail to spread modules in some conditions. Therefore, this paper proposes a new cosine-shaped function to resolve the problem. Our experimental results show that the cosine-shaped function can obtain a little better result than the bell-shaped function.

The characteristics of the papers include:

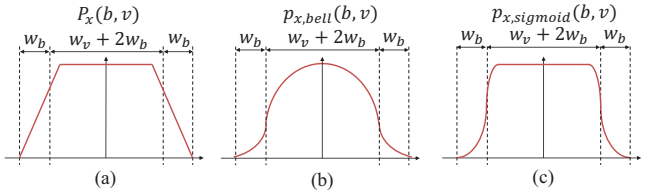

Fig. 2. (a) Original overlap function. (b) Bell-shaped function. (c) Sigmoid function.

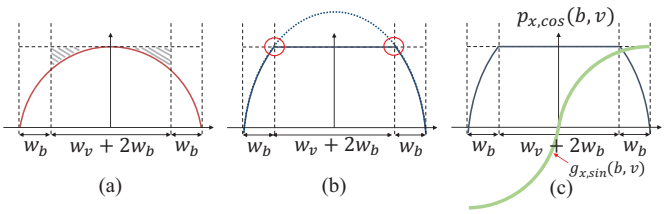

Fig. 3. (a)Original cosine-shaped function. (b)Modified cosine-shaped function. (c)The curve of the gradient function with respect to the curve of the utilization function.

- A general floorplanning methodology for 3D ICs with an arbitrary bonding style.
- Two layer assignment approaches to allocate modules to tiers for two kind bonding styles.
- A new cosine-shaped function to calculate utilization of modules in each bin.

II. Problem Formulation

Let $V = \{v_i | i \in \mathbb{Z}^+, 1 \leq i \leq n\}$ denote a set of modules, where n is number of modules. The width, height, and area of a module v_i are denoted by w_i, h_i, and a_i, respectively. The aspect ratio of a module v_i is defined as h_i/w_i. Given a 3D IC, its width, height, and number of tiers are determined, which are denoted by W_f, H_f, and K, respectively. For a soft module v_i, we also need to determine the resulting shape. Let (x_i, y_i) denote the lower-left coordinate of v_i. Our objective is to minimize total wirelength and number of TSVs under the fixed-outline constraints.

We have three assumptions in this paper. First, the tiers in the $K-1$-th and K-th layers are stacked in the F2F bonding style while other tiers are stacked in the F2B bonding style for a 3D IC with the hybrid bonding style. Hence, two models are used to estimate wirelength in the different bonding style. A 3D net is the net that passes through more than one tier. We divide a 3D net into several subnets according to tiers. HPWL of each subnet in each tier is calculated, respectively. Then, the values are summed up if two contiguous subnets are connected by a TSV. Otherwise, we will find the HPWL of the two subnets if they are connected by a via. Second, only one via (TSV) is allowed to be inserted between two contiguous tiers in the F2F (F2B) bonding style for a 3D net. Last, I/O bumps will be situated randomly below first tier.

III. Cosine-Shaped Function

The section illustrates a new function to calculate the utilization of modules in a bin. Let $p_{x,cos}(b,v)$ denote a potential value of a module v placed at a bin b in the x-axis. $p_{y,cos}(b,v)$ is defined, similarly. Without loss of generality, $p_{x,cos}(b,v)$ is shown in the following:

$$p_{x,cos}(b,v) = cos\frac{disX}{w_v + 4w_b}\pi, \tag{1}$$

where w_v is width of a module v, w_b is width of a bin b, $disX$ is the distance between a module v and a bin b. The denominator represents that the range of a module v which has impact on a bin b, and this is also the length of a half-period of the cosine function. Fig. 3(a) shows the curve of an original cosine shaped function. Compared to the original overlap function in Fig. 2(a), there still exists some error (see the shade regions in Fig. 3(a)). To resolve the problem, we scale the cosine function by multiplying a factor α, and the new function is shown in the following:

$$p_{x,cos}(b,v) = \alpha \times cos\frac{disX}{w_v + 4w_b}\pi, \tag{2}$$

Note that the variable α determines the amplitude of the function. Our target is to make the value of the cosine function equal to 1 when the value $|disX| \leq w_v/2 + w_b$ (the points are highlighted by the circles in Fig. 3(b)). Therefore, α is equal to the value in the following:

$$\alpha = \frac{1}{cos\frac{w_v+2w_b}{2w_v+8w_b}\pi}, \tag{3}$$

Finally, the value exceeding 1 is replaced by 1. Hence, the proposed cosine-shaped function $\hat{p}_{x,cos}(b,v)$ is shown in (4).

$$\hat{p}_{x,cos}(b,v) = \begin{cases} 1, & |disX| \leq \frac{w_v}{2} + w_b \\ \alpha cos\frac{disX}{w_v+4w_b}\pi, & \frac{w_v}{2} + w_b \leq |disX| \leq \frac{w_v}{2} + 2w_b \\ 0, & \frac{w_v}{2} + 2w_b \leq |disX| \end{cases} \tag{4}$$

See Fig. 3(c) for the shape of the proposed function.

The formulation of the analytical-based approach can be solved by the conjugate gradient algorithm, where the algorithm uses an iterative procedure to solve the non-linear formulation. In order to apply the algorithm, each term in the formulation has to be differentiable. The gradient of a utilization function represents the force and direction to push modules away from center of a placement region. However, the value is zero after the equation in (4) is differentiated when $|disX| \leq \frac{w_v}{2} + w_b$, which will cause the algorithm fail to spread modules in some conditions (note that a sigmoid function has the same problem). Hence, instead of using the modified cosine-shaped function, we take the original function $p_{x,cos}(b,v)$ ($p_{y,cos}(b,v)$) to obtain the the gradient function $g_{x,sin}(b,v)(g_{y,sin}(b,v))$ as follows:

$$g_{x,sin}(b,v) = \frac{\pi}{w_v + 4w_b} sin\frac{disX}{w_v + 4w_b}\pi, \tag{5}$$

The bold line in Fig. 3(c) represents the gradient function of the proposed cosine-shaped function. The sign of each point represents a moving direction.

IV. Our Methodology

Our methodology is composed of the partition-based layer assignment, global distribution, analytical-based layer assignment, legalization, and TSV assignment stages for a 3D IC with the hybrid bonding style. The top two tiers which are bonded in the F2F style is first considered as a combined tier. Hence, we apply a partition-based algorithm to allocate modules to K-1 tiers which are bonded by the F2B style, where K is number of tiers. Then, modules in all tiers are allocated to respective regions by an analytical-based approach. Next, the modules in the last partition are divided into two tiers which are bonded in the F2F style. Since rough locations of modules in the $K-1$-th tier have been determined in the previous stage, we apply another analytical-based algorithm to split the modules. Finally, the modules in each tier are legalized according to an ILP algorithm, and TSVs are inserted into empty space based on a network flow algorithm. These two stages are according to the procedures illustrated in SAINT [9]. Fig. 4 shows the flow of our methodology.

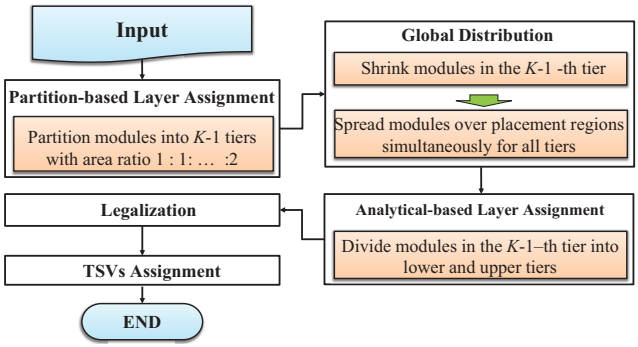

Fig. 4. Flow of our methodology.

A. Partition-based Layer Assignment

The subsection introduces a partition-based algorithm to perform layer assignment, which assigns modules to $K-1$ tiers, where the area ratio in each tier is 1: 1: ...: 2. The area in the $K-1$-th tier is twice larger than area in other tiers because we will further split the modules in the $K-1$-th tier in the later stage.

Algorithm 1 shows the pseudo code. We first apply hMetis [7] to partition a netlist into K parts (line 1). Since hMetis only minimizes cuts without considering area of TSVs, the area of modules in each partition may be unbalanced after TSVs are inserted. Hence, a target area for each partition which considers TSV area is predicted before we move modules in every neighboring modules. The target area in each tier is computed in line 2, where n denotes the number of modules; thus, $\sum_{i=1}^{n} a_i$ represents total module area. N_T is number of TSVs in current partition, and a_{TSV} denotes the area of a TSV. β is the ratio of additional TSVs which may be inserted with respect to N_T after modules are swapped in the later step (β is set to 1 in our experiment). Next, before we apply Fiduccia-Mattheyses (FM) algorithm to move modules in every two contiguous tiers from bottom to top (lines 4-13), modules in the $K-1$-th and K-th tiers are combined into one tier (see line 3). First, an initial gain of each module (i.e., implying change of cuts after a module is moved) is calculated. Let D_t (D_{t+1}) denote the t-th ($t+1$-th) tier. Then, we choose a module with the largest gain (line 7) and move the module to the other tier if the area constraint can be satisfied. The function $Move(m_i)$ in line 8 checks whether the target area constraint is satisfied after m_i is moved. Gain values are recalculated if the module is moved (see lines 9 - 10). The algorithm stops after $K-2$ iterations are performed.

Algorithm 1 Layer assignment flow for F2B bonding style.

Input: a netlist with modules and connection relationship
Output:assign modules to $K-1$ tiers with area ratio 1:1:...:2

1: Apply hMetis to get an initial partition of K parts
2: $Area_{target} = \frac{\sum_{i=1}^{n} a_i + (N_T(1+\beta\%)) \times a_{TSV}}{K}$
3: Combine $K-1$ and K tiers
4: **for** ($t = 1$; $t < K-1$; $t++$) **do**
5: Initial gains for modules in D_t and D_{t-1}
6: **repeat**
7: Pick m_i from D_t and D_{t-1} with Max gain
8: **if** $Move(m_i) ==$ TRUE **then**
9: move module to another tier
10: update gain
11: **end if**
12: **until** ($A_t \approx A_{t-1}$)
13: **end for**

B. Global Distribution Stage

After modules are assigned to $K-1$ tiers in the previous stage, the global distribution stage will spread modules to the placement region in each tier simultaneously. However, the area of modules in the $K-1$-th tier is twice larger than that in other tier, it is impossible to satisfy the utilization constraint in the analytical-based algorithm. Hence, the area of each module in the $K-1$-th tier is shrunk by half by dividing its height and width with $\sqrt{2}$ before the analytical-based algorithm is applied [9].

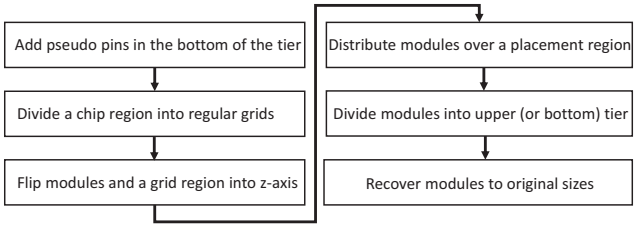

Fig. 5. Layer assignment for tiers in the F2F bonding style.

C. Analytical-based Layer Assignment

This subsection shows a procedure to divide modules in the $K-1$-th tier into two parts. Fig. 5 shows the steps of the procedure.

Since locations of modules in the lower tiers have been determined, we have to consider their locations when we divide modules in the $K-1$-th tier in order to obtain a better result. A chip region is first divided into regular grids with an identical size. Then, we perform layer assignment for the modules in each grid. Note that a grid size has to be careful determined because a utilization constraint is more easy to be satisfied if a grid size is larger. But a good placement result obtained in the previous stage may be violated. Next, modules and a placement region in each grid are flipped by 90 degree into z direction which is perpendicular to the $x-y$ plane, and they are redistributed over the region of a grid by applying an analytical-based algorithm. According their locations relative to the center of gravity of the modules, these modules are divided into two parts. Finally, modules are recovered to their original areas.

V. Experimental Results

We implemented our algorithm in C++ programming language and ran it on Linux workstation with Intel Xeon E5-2620 2.00 GHz CPU and 90GB memory. The ILP formulation was solved by CPLEX [14]. In all experiments, the aspect ratio of a soft module is restricted between 1/3 and 3. Whitespace in a tier is set to 15%, and the size of a TSV is 3μm $\times$ 3μm.

A. Experimental Results I

The first subsection demonstrates the feasibility of the proposed cosine-shaped function. We compare the consine-shaped function with the bell-shaped function in 2D floorplanning based on IBM benchmarks.

The comparison results are shown in Table I, where the aspect ratio of a chip outline is set to 1. The number of modules is shown in column 2. The results of bell-shaped function are listed in columns 2-3 while our results are shown in columns 4-5. The first column shows the names of the circuits, which also represents the numbers of modules in the circuits. Columns 2 and 4 show the wirelength while columns 3 and 5 show the runtime, respectively. The table shows that our cosine-shaped function can lead to better wirelength than the bell-shaped function by 1.4% to 2.3%. But our runtime is slightly longer than it. The experimental results also demonstrate the feasibility of the cosine-shaped function.

TABLE I
COMPARISONS OF TWO UTILIZATION FUNCTIONS IN 2D FLOORPLANNING.

Cir. (1/1)	# of Modules	Bell-shaped function		Cosine shaped function	
		WL(μm)	Time(s)	WL(μm)	Time(s)
ibm01	911	2794040	488	2717500	484
ibm02	1471	6164330	7230	6251920	2860
ibm03	1289	8778780	2021	8814740	3852
ibm04	1584	9402260	4636	9332140	3577
ibm06	749	8693820	255	8718500	59
ibm07	1120	15206500	816	15592400	1074
ibm08	1269	17255000	609	17039600	592
ibm09	1113	16129400	3044	14819100	1130
ibm10	1595	39366200	16960	37498700	21500
ibm11	1497	25596800	1750	25813600	4382
ibm12	1233	47979500	446	48078800	1178
ibm13	954	33529100	420	32310200	543
ibm14	1635	59490200	4078	59456200	4179
ibm15	1412	73154600	1469	74563100	761
ibm16	1091	92908800	359	91904100	491
ibm17	1442	134819000	6302	133948000	7188
ibm18	943	67806800	340	66693900	402
Nor.		1.008	0.93	1.000	1.000

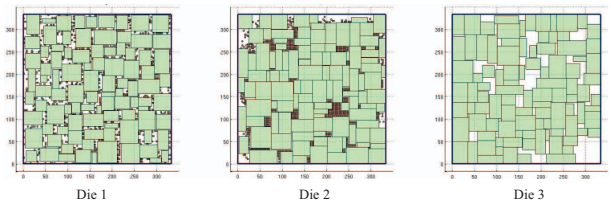

Fig. 6. Resulting floorplanns in a 3D IC for n300 with the hybrid bonding style.

B. Experimental Results II

This section compares our methodology with an analytical-based methodology proposed by Hsu *et al.* [2] in 3D floorplanning. We implemented the placement algorithm [2] by ourself and performed it on 3D floorplanning. The key difference between two methodologies is that Hsu *et al.* [2] use the analytical-based algorithm to allocate modules to different tiers as well as over placement regions in tiers simultaneously. However, we first use the partition-based approach to assign modules to different tiers and then use the analytical-based approach to distribute modules over placement regions.

Table II shows the comparisons between two methodologies on 3D ICs with the hybrid bonding style. The number of TSVs and wirelength are shown in columns 2 and 5 and columns 3 and 6, respectively. In order to connect modules to I/O bumps, elementary TSVs are required no matter which algorithm is applied. For clarity, the first value in the column shows the number of TSVs induced by different approach while the second value shows the number of TSVs required to connect modules to I/O bumps. The table shows that Hsu *et al.* [2]'s approach lead to significantly large number of TSVs since the objective function of its analytical-based approach has to consider wirelength, area balance, and number of TSVs at the same time while spreading modules over 3D space. Although the methodology by Hsu *et al.* [2] will reserve space during the global distribution stage for inserting TSVs in the later stage, they do not give initial locations of TSVs which make TSVs inserted at poor locations in legalization stage which result in longer wirelength. Fig. 6 displays the floorplanning results of n300 on a 3D IC using the hybrid bonding style.

VI. CONCLUSION

This paper has proposed a general floorplanning methodology which can be applied to 3D ICs with arbitrary bonding style. Besides, we have proposed a new cosine-shaped function to calculate utilization in the analytical-based approach. The experimental results have demonstrated the effectiveness and efficiency of our methodology.

TABLE II
COMPARISONS IN 3D FLOORPLANNING WITH THE HYBRID BONDING STYLE.

Cir.	[2]			Ours		
	#TSVs	WL 10^3 (μm)	Time (s)	#TSVs	WL 10^3 (μm)	Time (s)
n100	365(334)	139.2	3	288(334)	124.0	4.3(1.2)
n200	965(564)	243.3	9	637(564)	228.4	13.1(4.9)
n300	1120(569)	338.2	13	649(569)	31.7	20.7(7.5)
Nor.	1.29	1.08	0.65	1.00	1.00	1.00
2/1						
n100	460(334)	139.9	4	288(334)	133.8	5.21(1.2)
n200	1121(564)	265.0	12	637(564)	245.6	16.85(4.9)
n300	1286(569)	362.7	18	649(569)	337.9	26.1(7.5)
Nor.	1.42	1.07	0.7	1.00	1.00	1.00
3/1						
n100	455(334)	161.0	6	288(334)	145.4	7.31(1.2)
n200	1130(564)	289.3	16	637(564)	267.5	20.1(4.9)
n300	1217(569)	40.5	24	649(569)	367.1	33.5(7.5)
Nor.	1.4	1.09	0.74	1.00	1.00	1.00

REFERENCES

[1] E. Beyne, "The 3-D Interconnect Technology Landscape," *in IEEE MDAT*, pp. 8-20, 2014.

[2] M.-K. Hsu, V. Balabanov, and Y.-W. Chang, "TSV-Aware Analytical Placement for 3-D IC Designs Based on a Novel Weighted-Average Wirelength Model," *in IEEE TCAD*, vol. 32, no. 4, pp. 497-509, 2013.

[3] M.-K. Hsu, Y.-F. Chen, C.-C. Huang, S. Chou, T.-H. Lin, T.-C. Chen, and Y.-W. Chang, "NTUplace4h: A Novel Routability-Driven Placement Algorithm for Hierarchical Mixed-Size Circuit Designs," *in IEEE TCAD*, vol. 33, no. 12, pp. 1914-1927, 2014.

[4] M. Jung, T. Song, Y. Wan, Y.Peng, and S. K. Lim, "On Enhancing Power Benefits in 3D ICs:Block Folding and Bonding Styles Perspective," *in Proceedings of DAC*, pp. 1-6, 2014.

[5] A. B. Kahng and Q. Wang, "Implementation and Extensibility Of An Analytic Placer," *in IEEE TCAD*, vol. 24, no. 5, pp. 734-747, May 2005.

[6] J. Knechtel and Evangeline F. Y. Young, and J. Lienig, "Planning Massive Interconnects in 3D Chips." *in IEEE TCAD*, vol. 34, no. 11, pp. 1808-1821, 2015.

[7] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar, "Multilevel hypergraph partitioning: Application in VLSI domain," *in Proceedings of DAC*, pp. 526-529, 1997.

[8] C.-R. Li, W.-K. Mak, and T.-C. Wang, "Fast Fixed-Outline 3-D IC Floorplanning with TSV Co-Placement," *in IEEE TVLSI*, vol. 21, no. 3, pp. 523-532, 2013.

[9] J.-M. Lin, P.-Y. Chiu, and Y.-F. Chang, "SAINT: Handling Module Folding and Alignment in Fixed-outline Floorplans for 3D ICs," *in Proceedings of ICCAD*, pp. 1-7, 2016.

[10] J.-M. Lin, and J.-A. Yang, "Routability-driven TSV-aware Floorplanning Methodology for Fixed-outline 3D ICs," *to be appeared in IEEE TCAD*.

[11] T. Song, S. Panth, and Y.-J. Chae, "Three-tier 3D ICs for more power reduction: Strategies in CAD, design, and bonding selection," *in Proceedings of ICCAD*, pp. 752-757, 2015.

[12] M.-C Tsai, T.-C. Wang, and T.-T. Hwang,"Through-Silicon Via Planning in 3-D Floorplanning." *in IEEE TVLSI*, vol. 19, no. 8, pp. 1448-1457, Aug 2011.

[13] P.-H. Yuh, C.-L. Yang, Y.-W. Chang, and H.-L. Chen, "Temporal Floorplanning Using 3D-subTCG," *in Proceedings of ASP-DAC*, pp. 725-730, 2004.

[14] [Online]Available:http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/