

# Wirelength And Routability For Fixed Outline Networks

<sup>[1]</sup> S.P.Kalaiarasi, <sup>[2]</sup> B.Srinath, <sup>[3]</sup> P.Arunapriya  
<sup>[1][2][3]</sup> SRM University

**Abstract-** In this paper, we propose an SKB-tree representation for the separation of multiple supply voltage (MSV) of modules in Integrated Circuits(IC) and routability at the same time under the fixed-outline constraint. Apart from previous works, we constrain modules of the same voltage to be placed into one region for wirelength optimization. This proposed methodology results, can reduce wirelength and be routing congestion in ICs. Our approach guarantees to obtain the minimum wirelength in time. The algorithm finds the position of modules in ICs to reduce the wirelength. It will dynamically allocate modules in fixed outline Integrated Circuits. This algorithm is implemented in GSRC bench circuits for wirelength optimization.

**Index Terms**— Fixed-outline, multiple supply voltage, voltage island, wirelength.

## I. INTRODUCTION

Integrated circuits (ICs) are tied to our daily life in a pervasive manner. In modern IC design, handling multiple supply voltage modules and reducing the wirelength having more complexity. Interconnect has become critical determiner of circuit performance. Circuit placement is starting to play an important role in today’s high-performance IC designs. In addition to wirelength optimization, the issue of reducing excessive congestion in local regions such that the router can finish the routing successfully is become important problem. Some modules may be safely abutted for the benefit of reduction in area and wirelength. Some other may be placed with extra whitespace in-between. Wirelength reduction has an impact on the cycle time and energy dissipation.

Multiple Supply Voltage (MSV) is a technique that focuses on reducing the dynamic power. MSV introduce some difference on traditional physical synthesis due to the different voltage operations of each region in the design. MSV technique partitions the internal logic of a chip into multiple voltage regions such that each part has its own supply voltage. Low power has become a burning issue in modern IC design. Based on the similar voltage the modules are placed together.

### (a) Voltage Island Driven Representation:

SKB tree is used for voltage-island driven floorplanning for placing the modules which have same voltage into fixed-outline Island, which can able to reduce power and wirelength. SKB-Tree also called as a left-right skewed B\*-tree. The figure 1 shown below is the SKB tree with levels of the tree as Voltage islands. Compare to the B\*-Tree [3] it searching time is low because it has only

right-Childs. It does not need contour during the packing of the modules.

A zero dead space with the modules have achieved very easily by using the SKB-Tree representation. The same voltage is placed in island which makes optimization of area and wire length difficult. To solve this block voltage determined using dynamic programming to obtain optimal power consumption and initialize Voltage Island based on the result.



Fig 1: Floorplan Representation using SKB Tree

**International Journal of Engineering Research in Electronics and Communication  
Engineering (IJERECE)  
Vol 5, Issue 5, May 2018**

**(b)Our Contribution:**

In this paper, SKB tree representation is used to separate the modules by voltage island constraints. Our proposed algorithm is used to find the optimal dimensions between the modules. The optimal dimension can be calculated based on the width and height of the modules without changing the area. Minimum value of optimal dimension should be taken. The midpoint of each module can be calculated. Each module can be connected based on the midpoint values. Based on the midpoint the wirelength between modules is calculated.

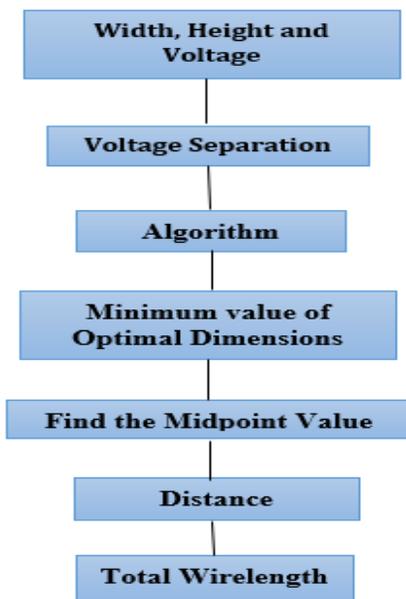


Figure 2: Proposed Design Flow

**II SKB TREE REPRESENTATION:**

SKB tree is used for voltage-island driven floorplanning for placing the modules which have same voltage into fixed-outline Island, which can able to reduce power and wirelength. SKB-Tree also called as a left-right skewed B\*-tree. The figure 1 shown below is the SKB tree with levels of the tree as Voltage islands. Compare to the B\*-Tree [3] it searching time is low because it has only right-Childs. It does not need contour during the packing of the modules. A zero dead space with the modules have achieved very easily by using the SKB-Tree representation.

The same voltage is placed in island which makes optimization of area and wire length difficult. To solve this block voltage determined using dynamic programming to

obtain optimal power consumption and initialize Voltage Island based on the result.

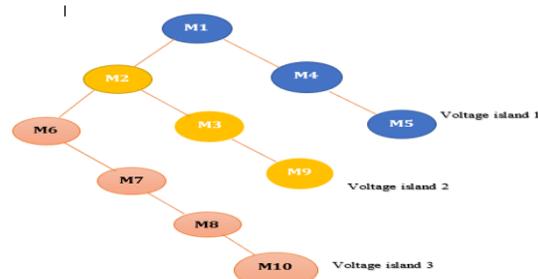


Fig 3: SKB Tree

**III RELATED WORK:**

The simulated annealing approach is used to which partitioned blocks into several, islands both in chip. To simultaneously consider of floorplanning and power saving, Hu et al. [1] used a graph to represent a partition solution of voltage islands and iteratively perturb the graph until the final solution is obtained. Hung et al. [2] the simulated annealing framework is limited for runtime for increase significantly. The number of blocks in a single chip increase dramatically.

Lee et al. [3] has approached for power saving and timing constraint and is mainly used to assign the blocks in voltage. It considered about level shifter and power network. Different consideration is used to deal with voltage island generation by applying the dynamic programming. Then, Ma et al. [4] used a normalized polish expression (NPE) to deal with this problem. In their method, they wished to do voltage assignment, voltage island generation, and floorplanning in the same time. However, because power consumption, total wirelength, and chip area are optimized at the same time in the simulated annealing, it is not easy to get good results for all these factors in the simulated annealing framework. In a real design, the number of voltages that can be realizable in a single chip may be smaller than the number of supply voltages provided. Although more supply voltages could reduce the power consumption of a chip, additional design complexity induced by it may offset the power and area savings [5]Sengupta and Saleh [6] On the Power State Model and connectivity information is used for the dynamic algorithm to build a voltage assignment table for blocks and partitioned blocks into islands. In this approach it gives the optimal solution for voltage selection but this method has time complexity Liu et al.'s [7]. Lot of redundancy is

**International Journal of Engineering Research in Electronics and Communication  
Engineering (IJERECE)  
Vol 5, Issue 5, May 2018**

induces in true 3-D representation based on this space is issue in additional data structure. [8]. To construct a floorplans in different device layers the array of an existing 2-D representation is used.[9]. These approaches first allocate modules into different layers and then apply intralayer operations and interlayer operations to perturb solutions within a layer as well as across different layers, [10] respectively. In addition to using the SA-based approach, some researches use different approaches to deal with 3-D floorplanning such as 3-D-STAF [11] which applied the force-directed technique to distribute modules to 3-D space. Besides, Li et al. [12] used the generalized slicing tree (GST) with the enumerative packing (EP) [13] to legalize modules and TSVs in a fixed outline after modules have been allocated to different tiers. Li et al. [14] has approached is used to require a longer runtime to get a good solution, for this SA-based approaches algorithm is used. It presented an efficient and effective 3-D fixed-outline floorplanner, named Co-place, to handle the problem. Yan and Chu [15], is constructed for each tier by applying the round-robin hypergraph partitioning algorithm, and feasible floorplans of all tiers are obtained by the GSTs. Finally, they arrange TSVs by the network-flow-based algorithm which further optimizes total wirelength. Li et al. [16] proposed to use probabilistic analysis to estimate the routing congestion for 3-D ICs. To make the result more accuracy, the influence of TSV locations is also considered in the model. Recently, Panth et al. [17] discussed the routability-driven placement for standard cells in monolithic 3-D ICs. They developed a routing demand model for monolithic 3-D-ICs, and used it to develop a min-overflow practitioner which is able to enhance routability by off-loading demand from one tier to another.

**Problem formulation:**

In this project, we propose an algorithm to find an optimal dimension for reducing the wirelength between the modules. Skewed binary tree (SKB) is used to place the modules have same voltage into fixed-outline.

Let  $M = \{m_1, m_2, \dots, m_n\}$  be a set of  $n$  modules whose width, height, and area. The modules have a variable width and height with a fixed area. Algorithm is used to find the sequence of modules. To find an optimal dimension the values of width and height are interchanged. Then based on the interchanged value the minimum value should be taken.

For the purpose of reducing the wirelength and to satisfy the voltage island constraint, we propose an algorithm to determine optimal dimensions of the modules. SKB tree is used for positioning the modules in their respective voltage islands. According to its result, our proposed algorithm can initialize a voltage island for each supply voltage and

reduces the wirelength in the design. Comparison of existing SKB methodology the run time and wirelength are reduced.

**IV PROPOSED ALGORITHM:**

```

1:  $S = \emptyset; P = \emptyset;$ 
2: top:
3: while  $m_i, m_j$  do ;
4:      $(u, v) = [w_i + w_j, \max\{h_i, h_j\}];$ 
5:      $S+ = (u, v);$ 
6:     if  $u > v$  then
7:          $(u, v) = [t_1(h_i) + h_j, \max\{t_1(w_i), h_j\}];$ 
8:          $S+ = (u, v);$ 
9:     else
10:         $(u, v) = [t_2(h_j) + h_i, \max\{t_2(w_j), h_i\}];$ 
11:         $S+ = (u, v);$ 
12:    end if
13: end while
14: find minimum of S;
15: while  $m_i, m_j$  do
16:     $(u, v) = [\max\{m_i(w_i), m_j(w_j)\}, m_i(h_i) + m_j(h_j)];$ 
17:     $P+ = (u, v);$ 
18:    if  $(u > v)$  then
19:         $(u, v) = [\max\{t_1(h_i), m_j(h_j)\}, t_1(w_i) + m_j(h_j)];$ 
20:         $P+ = (u, v);$ 
21:    else
22:         $(u, v) = [\max\{t_2(h_j), m_i(h_i)\}, t_2(w_j) + m_i(h_i)];$ 
23:         $P+ = (u, v);$ 
24:    end if
25: end while
26: find minimum of P;
27: find minimum (S,P);
28: update minimum area dimension to  $m_i, m_j;$ 
29:  $i \leftarrow i + 1;$ 
30:  $j \leftarrow j + 1;$ 
31: goto top;
    
```

Using this algorithm, we identify the optimal dimensions.

Consider the dimensions  $(w_i; h_i)$  and  $(w_j; h_j)$  as modules  $m_i$  and  $m_j$ . Let  $t_1$  be the duplicate dimension of module  $m_i$  whose width  $w_i = h_i$  and  $h_i = w_i$ . Similarly, assume  $t_2$  for module  $m_j$  with width  $w_j = h_j$  and  $h_j = w_j$ . We compute the resulting dimension after merging modules A and B using the equation 4.1,

$$(u, v) = [w_i + w_j, \max\{h_i, h_j\}]$$

**International Journal of Engineering Research in Electronics and Communication  
Engineering (IJERECE)  
Vol 5, Issue 5, May 2018**

where  $u$  contains the resulting width and  $v$  contains the resulting height. If the value of  $u$  is greater than  $v$ , we use Equation 4.2,

$$(u, v) = [t1(hi) + hj, \max\{t1(wi), hj\}]$$

For  $u$  less than  $v$ , we use Equation 4.3 to compute the new dimensions of  $u$  and  $v$ .

$$(u, v) = [t2(hj) + hi; \max\{t2(wj), hi\}];$$

The resulting dimensions from Equations 4.1, 4.2, and 4.3 are stored in a set  $S$ . In the same way, in step (16–26), we use the following equations (4.4-4.6) to find height-wise optimal dimensions.

$$(u; v) = [\max\{wi;wj\}, hi + hj]$$

$$(u; v) = [\max\{t1(hi); hj\}, t1(wi) + hj]$$

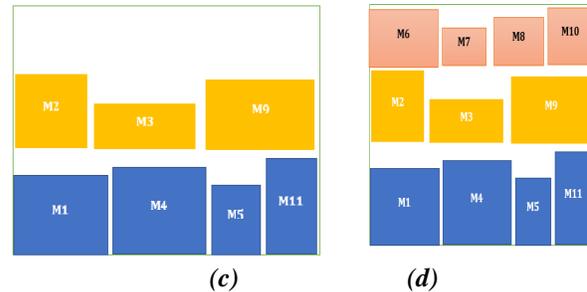
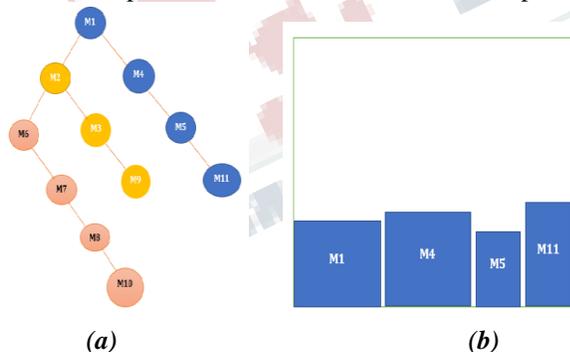
$$(u; v) = [\max\{t2(hj); hi\}, t2(wj) + hi]$$

The resulting dimensions from Equation 4.4, 4.5, and 4.6 are stored in a set  $P$ .

A minimal area dimension area is chosen from the set  $S$  and  $P$  and we update the dimensions of modules  $mi$  and  $mj$ .

**(a) Placement:**

Based on the SKB tree representation, the modules are separate by the Voltage Island. After that Using this algorithm, we obtain an optimal dimension of the module. Based on the optimal dimension of the modules are placed.



**Fig 4: Placement Process after Implementation of Proposed Algorithm**

**(b) Algorithm Explanation:**

To find the optimal dimensions of the module the inputs are given,

No	width	height	voltage
1	3	2	v3
2	5	3	v2
3	6	1	v2
4	3	5	v1

**Table 1 Input Values**

After applying the SKB algorithm the voltage are arranged based on the voltage order,

No	width	height	voltage
1	3	5	v1
2	5	3	v2
3	4	1	v2
4	3	2	v3

**Table 2 Inputs Value are Arranged by Voltage**

The algorithm proceeds with the following steps given below.

The dimension of module  $m1$  and  $m2$  are taken.

For the vertical orientation, the dimensions are  $u = (3, 5)$  and  $v = (5, 3)$

(i)  $(u, v) = \{w1 + w2, \max(h1, h2)\} = \{(3 + 5), \max(5, 3)\} = (8, 5) = 40.$

(ii) Interchange the value of module  $m1$  as  $w = h$  and  $h = w$ , then the dimensions are  $u = (5,3)$  and  $v = (5,3).$

$(u, v) = \{w1 + w2, \max(h1, h2)\} = \{(5 + 5), \max(3, 3)\} = (10, 5) = 50.$

For the horizontal orientation, the dimensions are  $u = (3, 5)$  and  $v = (5, 3)$

**International Journal of Engineering Research in Electronics and Communication  
Engineering (IJERECE)  
Vol 5, Issue 5, May 2018**

- (i)  $(u, v) = \{ \max(w_1, w_2), (h_1 + h_2) \} = \{ \max(3, 5), (5 + 3) \} = (5, 8) = 40.$
- (ii) Interchange the value of module  $m_2$  as  $w = h$  and  $h = w$ , then the dimensions are  $u = (5, 3)$  and  $v = (5, 3).$   
 $(u, v) = \{ \max(w_1, w_2), (h_1 + h_2) \} = \{ \max(5, 5), (3 + 3) \} = (5, 6) = 30.$

Based on the above calculation, minimum value should be taken. The dimensions are  $u = (5, 3)$  and  $v = (5, 3).$

$$N1 = (5, 3) (5, 3) = (5, 6) = 30.11$$

Next calculation is based on the new dimension  $N1$  and  $m_3.$  The dimension of module  $N1$  and  $m_2$  are taken.

For the vertical orientation, the dimensions are  $u = (5, 6)$  and  $v = (4, 1)$

- (iii)  $(u, v) = \{ w_1 + w_2, \max(h_1, h_2) \} = \{ (5 + 4), \max(6, 1) \} = (9, 6) = 45.$
- (iv) Interchange the value of module  $m_1$  as  $w = h$  and  $h = w$ , then the dimensions are  $u = (6, 5)$  and  $v = (4, 1).$   
 $(u, v) = \{ w_1 + w_2, \max(h_1, h_2) \} = \{ (6 + 4), \max(5, 1) \} = (10, 5) = 50.$

For the horizontal orientation, the dimensions are  $u = (5, 6)$  and  $v = (4, 1)$

- (iii)  $(u, v) = \{ \max(w_1, w_2), (h_1 + h_2) \} = \{ \max(5, 4), (6 + 1) \} = (5, 7) = 35.$
- (iv) Interchange the value of module  $m_2$  as  $w = h$  and  $h = w$ , then the dimensions are  $u = (5, 6)$  and  $v = (1, 4).$   
 $(u, v) = \{ \max(w_1, w_2), (h_1 + h_2) \} = \{ \max(5, 1), (6 + 4) \} = (5, 10) = 50.$

Based on the above calculation, minimum value should be taken. The dimensions are  $u = (5, 6)$  and  $v = (4, 1).$

$$N1 = (5, 6) (4, 1) = (5, 7) = 35.$$

Based on the above procedure further calculation are done for the remaining modules.

The dimension values are taken as a coordinates. The coordinates are

$x_1$	$x_2$	$y_1$	$y_2$
0	3	0	5
0	9	6	6
6	10	6	7
0	3	10	12

**Table 4.3: coordinates of the dimensions**

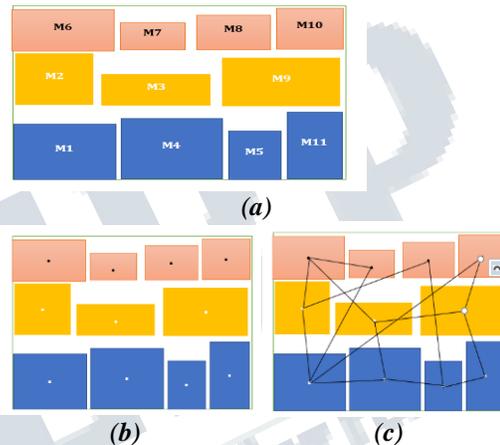
Each module can be connected based on the midpoint of the module. The below formula is used to find the midpoint of each module.

$$\text{Midpoint} = \left( \frac{x_1 + x_2}{2}, \frac{y_1 + y_2}{2} \right)$$

Based on the midpoint value, the each module is interconnected together and finds the distance.

$$\text{Distance} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

After the distance calculation of the total distance are calculated.



**Fig 5: Placing the Modules in IC, Midpoint and Distance**

**V EXPERIMENT RESULTS**

In this section, we present two set of experiment for voltage-island driven using SKB tree representation and reducing the wirelength. Our programs were implemented in the java programming language and compiled by eclipse.

**(a) Comparison of Voltage Island Representation:**

We first compare our algorithm with all the state-of-the-art SKB tree representation. We only show the results of the circuits  $n_{50}$ ,  $n_{100}$  and  $n_{300}$ . These experiments were performed on GRSC benchmark. In the experiment, we try to perform SKB tree representation method to separate the voltage-island and our algorithm is used to find the optimal dimensions to reduce the wirelength. Using our algorithm we achieved the better results of wirelength. In the voltage island representation used more voltages and circuits. In this method, we used only three voltages and three circuits. Comparing to the pervious method we achieved the better results and reduced the wirelength.

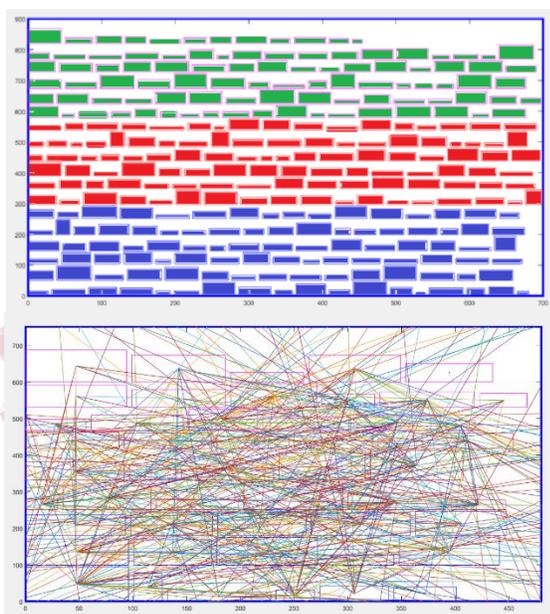
**International Journal of Engineering Research in Electronics and Communication  
Engineering (IJERECE)  
Vol 5, Issue 5, May 2018**

**(b) Comparison of SKB Tree Representation:**

In the second set of experiments, we show the results of SKB tree representation. We only compared with our algorithm with SKB tree. Compared to existing approach we achieved the better results. Runtime is also decreased. Comparison table is shown below:

modules	voltage island-driven floorplaning		SKB representation		Ours	
	wirelength	runtime	Wire length	Run time	Wire length	Run Time
n50	83899	93.0	76783	43	182360	23.4s
n100	138550	377.6	121637	83	312450	25.5s
n300	499921	3156.3	410570	2327	133930	27.8s

*Table 4: Comparison between the existing method and proposed method*



(a) (b)

*Fig 6: After implementation of proposed algorithm*

## VI CONCLUSION

In this paper, we have proposed an SKB tree representation for separation of voltage in fixed outline IC. In the project, we concentrate two things without changing the area of the module, find the optimal dimension of the module and modules are separate by voltage to avoid the delay. In addition, our algorithm is used to find the optimal dimensions to reduce the wirelength, it achieves practically

with GRSC Benchmark. The experimental results have demonstrated that our algorithm can achieve the reduction in wirelength

## REFERENCES

- [1] J. Hu, Y. Shin, N. Dhanwada, and R. Marculescu, "Architecting voltage islands in core-based system-on-a-chip designs," in Proc. ISLPED, 2004, pp. 180–185.
- [2] W.-L. Hung, G. M. Link, Y. Xie, N. Vijaykrishnan, N. Dhanwada, and J. Conner, "Temperature-aware voltage islands architecting in system-on-chip design," in Proc. ICCD, 2005, pp. 689–696.
- [3] D. E. Lackey, P. S. Zuchowski, T. R. Bednar, D.W. Stout, S. W. Gould, and J. M. Cohn, "Managing power and performance for system-on-chip designs using voltage islands," in Proc. ICCAD, 2002, pp. 195–202.
- [4] W.-P. Lee, H.-Y. Liu, and Y.-W. Chang, "Voltage island aware floorplaning for power and timing optimization," in Proc. ICCAD, 2006, pp. 389–394.
- [5] Q. Ma and E. F. Y. Young, "Multivoltage floorplan design," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 29, no. 4, pp. 607–617, Apr. 2010.
- [6] S. Nakatake, K. Fujiyoshi, H. Murata, and Y. Kajitani, "Module placement on BSG-structure and IC layout applications," in Proc. ICCAD, 1996, pp. 484–491.
- [7] D. Sengupta and R. Saleh, "Application-driven floorplan-aware voltage island design," in Proc. DAC, 2008, pp. 155–160.
- [8] L. Cheng, L. Deng, and M. D. F. Wong, "Floorplaning for 3-D VLSI design," in Proc. ASP-DAC, Shanghai, China, Jan. 2005, pp. 405–411.
- [9] R. Fischbach, J. Lienig, and J. Knechtel, "Investigating modern layout representations for improved 3D design automation," in Proc. GVLSI, Lausanne, Switzerland, May 2011, pp. 337–342.
- [10] J. Cong, J. Wei, and Y. Zhang, "A thermal-driven floorplaning algorithm for 3D ICs," in Proc. ICCAD, San Jose, CA, USA, Nov. 2014, pp. 306–313.
- [11] J. Knechtel, E. F. Y. Young, and J. Lienig, "Planning massive interconnects in 3-D chips," IEEE Trans. Comput.-

**International Journal of Engineering Research in Electronics and Communication  
Engineering (IJERECE)  
Vol 5, Issue 5, May 2018**

---

Aided Design Integr. Circuits Syst., vol. 34, no. 11, pp. 1808–1821, Nov. 2015.

[12] P. Zhou et al., “3D-STAF: Scalable temperature and leakage aware floorplanning for three-dimensional integrated circuits,” in Proc. ICCAD, San Jose, CA, USA, Nov. 2007, pp. 590–597.

[13] C.-R. Li, W.-K. Mak, and T.-C. Wang, “Fast fixed-outline 3-D IC floorplanning with TSV co-placement,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 3, pp. 523–532, Mar. 2013.

[14] J. Z. Yan and C. Chu, “DeFer: Deferred decision making enabled fixedoutline floorplanning algorithm,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 3, pp. 367–381, Mar. 2010.

[15] S. Chen and T. Yoshimura, “Multi-layer floorplanning for stacked ICs: Configuration number and fixed-outline constraints,” Integr. VLSI J., vol. 43, no. 4, pp. 378–388, Sep. 2010.

[16] C.-R. Li, W.-K. Mak, and T.-C. Wang, “Fast fixed-outline 3-D IC floorplanning with TSV co-placement,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 21, no. 3, pp. 523–532, Mar. 2013.

[17] J. Z. Yan and C. Chu, “DeFer: Deferred decision making enabled fixedoutline floorplanning algorithm,” IEEE Trans. Comput.-Aided Design Integr. Circuits Syst., vol. 29, no. 3, pp. 367–381, Mar. 2010.

[18] W. Li, J. Kim, and J.-W. Chong, “A novel congestion estimation model and congestion aware floorplan for 3D ICs,” in Proc. ICIMTR, Malacca, Malaysia, May 2012, pp. 199–204.

[19] S. Panth, K. Samadi, Y. Du, and S.-K. Lim, “Placement-driven partitioning for congestion mitigation in monolithic 3D IC designs,” in Proc. ISPD, Petaluma, CA, USA, Jan. 2014, pp. 47–54.