Thermal-Aware Floorplanning with Min-cut Die Partition for 3D ICs

Cheoljon Jang and Jong-wha Chong

Three-dimensional integrated circuits (3D ICs) implement heterogeneous systems in the same platform by stacking several planar chips vertically with through-silicon via (TSV) technology. 3D ICs have some advantages, including shorter interconnect lengths, higher integration density, and improved performance. Thermal-aware design would enhance the reliability and performance of the interconnects and devices. In this paper, we propose thermal-aware floorplanning with min-cut die partitioning for 3D ICs. The proposed min-cut die partition methodology minimizes the number of connections between partitions based on the min-cut theorem and minimizes the number of TSVs by considering a complementary set from the set of connections between two partitions when assigning the partitions to dies. Also, thermal-aware floorplanning methodology ensures a more even power distribution in the dies and reduces the peak temperature of the chip. The simulation results show that the proposed methodologies reduced the number of TSVs and the peak temperature effectively while also reducing the run-time.

Keywords: Three-dimensional integrated circuit, die partition, floorplanning, physical design, VLSI.

Manuscript received Nov. 21, 2013; revised Jan. 23, 2014; accepted Feb. 3, 2014.

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2013-H0301-13-1011).

Cheoljon Jang (jangcj@hanyang.ac.kr) is with the Department of Nanoscale Semiconductor Engineering, Hanyang University, Seoul, Rep. of Korea.

Jong-wha Chong (corresponding author, jchong@hanyang.ac.kr) is with the Department of Electronics Computer Engineering, Hanyang University, Seoul, Rep. of Korea.

I. Introduction

The recent growths in semiconductor fabrication have enabled the use of three-dimensional integrated circuits (3D ICs). 3D ICs implement heterogeneous systems in the same platform by stacking several planar chips vertically [1]. Through-silicon vias (TSVs) are used for vertical inter-die connections in 3D ICs [2]. 3D ICs have some advantages, including shorter interconnect lengths, greater integration density, and improved performance [3]. However, the reduction in chip size of the 3D implementation induces a higher power density than that in 2D circuits. Due to the multiple heat sources in multiple dies, on-chip temperature increases rapidly as the die number increases [4]. The higher temperature causes a higher resistance value and a greater IR drop [5]. It eventually degrades the reliability and performance of interconnects and devices [6]. Therefore, a thermal-aware 3D IC design methodology is needed. Partitioning and floorplanning stages in the physical design of 3D ICs are important to improve results because they impact upon subsequent stages, including placement and routing. Partitioning is a process of dividing a chip into small blocks that can be designed or analyzed individually. Since interactions with external signals can only be performed through the bottom die, the partitioning problem for 3D ICs differs from that of the conventional 2D ICs. For 3D ICs, a net that has been cut may contribute more than one TSV when its related partitions are far away from other partitions. References [7] and [8] propose multiway partitioning algorithms for conventional 2D ICs but do not guarantee an optimal outcome for 3D ICs. The authors in [9] propose integer linear programs for 3D IC partitioning problems. However, these programs have lots of computational complexity and as such need to be

© 2014 ETRI

improved upon. Floorplanning is a process of determining the shape and arrangement of subcircuits or modules. Reference [10] proposes a thermal-driven floorplanning algorithm for 3D ICs, which uses simulated annealing with an integrated compact thermal model. This simulated annealing-based stochastic optimization technique has long run-times that scale poorly with problem size. The authors in [11] and [12] propose a temperature-aware and force-directed floorplanning algorithm for 3D ICs. The solution of the force-directed floorplanning algorithms typically overlaps the modules that need to be resolved. Because a large number of TSVs also increase the on-chip temperature, initial physical design stages, including partitioning and floorplanning, are required to minimize the number of TSVs and reduce the peak on-chip temperature with a short run-time.

This paper proposes a die partition methodology based on the min-cut theorem that minimizes the number of TSVs in 3D ICs with multiple dies and a thermal-aware floorplanning methodology that reduces the peak on-chip temperature.

II. Problem Definition

1. Die Partition

The partitioning stage in the physical design of 3D ICs is called the die partition stage because smaller blocks divided from a design are assigned to each die. The die partition of a 3D IC with l dies is the process equivalent to partitioning the design into l dies using the minimum number of TSVs and the minimum footprint. TSV is a vertical interconnection inserted between dies. It serves to connect dies and to help deliver signals originating from the bottom die. Several signals can be delivered smoothly and properly to the upper dies when the number of TSVs is increased. However, because a TSV is comprised of a conductor crossing the Si substrate of the stacked dies, it can be modeled as resistance. If the number of TSVs is increased between dies, then the values of resistance and on-chip temperature are subsequently increased. Therefore, it would be better to minimize the number of TSVs to within a range that allows for the sufficient delivery of several signals. Given the number of dies and a design with logic blocks, I/O terminals, and the connectivity between them, the goal of min-cut die partitioning for 3D ICs is to find l partitions with the minimum number of TSVs and with the best area balance among partitions.

2. Floorplanning

The thermal-aware floorplanning problem for 3D ICs consists of a set of blocks $\{b_1, b_2, \ldots, b_n\}$. A block b_i is a rectangular module whose width, height, and power consumption are denoted by w_i, h_i, and $Power_i$, respectively. There is a fixed number of dies, Z. Each block is free to rotate. Let (x_i, y_i, z_i) denote the coordinates of the center of block b_i on a chip, where $1 \le z_i \le Z$. A floorplan F is an assignment of (x_i, y_i, z_i) for each b_i, $1 \le i \le m$, such that no two blocks overlap with each other. The goals of thermal-aware floorplanning for 3D ICs are to minimize (a) the number of TSVs, (b) peak on-chip temperature, (c) wirelength, and (d) chip area. Wirelength is the half-parameter wire length estimation. Chip area is the product of the maximal height and maximal width over all dies. The on-chip temperature is influenced by the density of power consumptions of the blocks in the chip. The power density increases because multiple dies are thinned and bonded together in a small space. As a result, a large amount of heat is generated in this small volume. The blocks with high power consumption increase in temperature more than those with low power consumption; therefore, in this paper, blocks with high power consumption are called hot blocks, while blocks with low power consumption are called cool blocks.

III. Min-cut Die Partition for Multiple Dies

The proposed min-cut die partition algorithm is largely divided into five steps, as shown in Fig. 1.

1. Generating Partitions Randomly

As the first step for the min-cut die partition, n partitions (which become n dies) are randomly generated. Because n partitions become n dies at the end of the algorithm, to balance the footprint between each die, the sum of the total area of blocks in each partition should satisfy the following area constraint:

$$\frac{\sum \text{Area}_i}{n+1} \le (\text{Area constraint}) \le \frac{\sum \text{Area}_i}{n-1}, \qquad (1)$$

where n and Area_i denote the number of partitions and the area of the ith block, respectively. The n partitions are filled with the

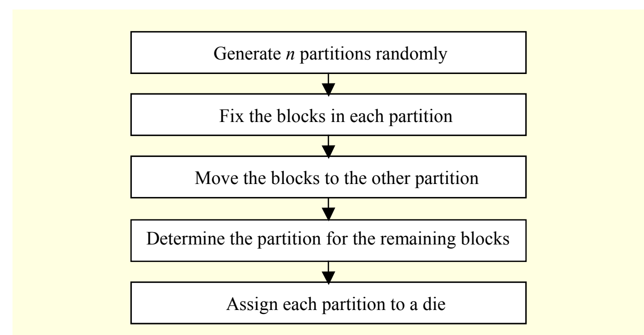

Fig. 1. Flow chart of proposed min-cut die partition algorithm.

http://dx.doi.org/10.4218/etrij.14.0113.1204

blocks and generated one by one according to the area constraint.

2. Fix Blocks in Each Partition

Among the blocks in each partition, each block is judged to determine whether keeping the block in the partition induces the min-cut result. If this is the case, then the block is fixed inside the partition. To obtain the min-cut result as the die partition result means that the connections among the dies are minimized and the number of TSVs to connect all dies becomes minimized. The following connective degree constraint is considered to fix the blocks:

$$\mathrm{Max}(\mathrm{E}_j(i)) \le \mathrm{E}(i)\,, \tag{2}$$

where $\mathrm{E}_j(i)$ represents a set of i incident edges connected with a jth partition not including block i, and $\mathrm{E}(i)$ represents a set of i incident edges connected with the partition, including block i. If the right side of (2) is greater than or equal to the left side, then the number of edges connected with the partition, including block, i is the greatest. This means that the number of TSVs can be minimized when block i is placed in the partition.

3. Move Blocks to Other Partitions

Each non-fixed block is judged in order of connectivity (from high to low) to determine whether moving the block to another partition induces the min-cut result. If this is the case, then the block is moved to the other partition. The number of TSVs is minimized when the connective degree constraint is satisfied and the number of uncut edges becomes largest after the block is moved to the specific partition. To prevent lots of blocks from gathering in a specific partition, the movement of blocks is only allowed within the range satisfying the area constraint, as in (1).

4. Determine Partition for Remaining Blocks

The remaining blocks (except those that are now fixed or that have been moved) are judged for assignment to a partition. Considering area balance among the partitions as a priority, when the number of uncut edges becomes largest while satisfying the area constraint after the block moves to another partition, the block is allocated to the partition. Naturally, when the area constraint is unsatisfied, the block is considered to be allocated to another partition.

5. Assign Each Partition to a Die

The number of TSVs differs depending on how the partitions are sorted when assigning the partitions to dies. Therefore, the partitions are assigned to dies for minimizing the number of TSVs. To further minimize the number of TSVs, it is advantageous to place two partitions with many interconnections close together. Because we only know the number of TSVs through the alignment of the partitions, which partition will be the top die or the bottom die is not a major consideration in this situation. For example, when there are three partitions and the numbers of connections between the partitions are given, there are three cases for assigning these three partitions to dies. The case with the minimum number of TSVs is the case that the partition with the most connections between other partitions is located at the center. Thinking inversely and in terms of a complementary set, the equivalent case is where two neighboring partitions having the fewest connections are subtracted from the total candidates.

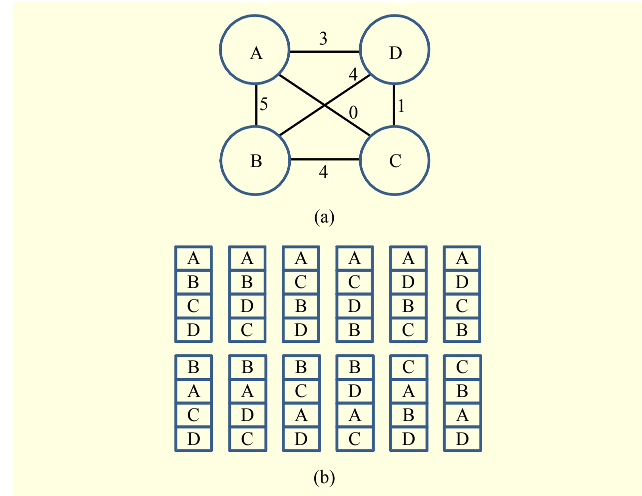

Fig. 2. Example of die partition with four partitions: (a) four partitions and the connections between them and (b) total candidates for die partition with four partitions.

For a more expanded example, when there are four partitions and the number of connections between each partition is as shown in Fig. 2(a), there are twelve cases for assigning these four partitions to dies. The sorted orders (A-B-C-D) and (D-C-B-A) indicate that partition D is assigned to the bottom die and that partition A is assigned to the bottom die, respectively. Which of these orders should be assigned to the bottom die or top die is determined not by the number of connections but by the power consumption of each die, so (A-B-C-D) and (D-C-B-A) are the same case. Therefore, there are twelve cases for assigning these four partitions to dies, as shown in Fig. 2(b). For searching the best die partition case with the minimized number of TSVs, the approach that subtracts the candidates of two neighboring partitions with the fewest connections from the total candidates is applied. This approach eliminates the candidates for the best case. In Figs. 2(a) and 2(b), the candidates that neighboring partitions A and C have the fewest connections between them are (A-C-B-D), (A-C-D-B), (B-A-

C-D), (B-C-A-D), (B-D-A-C), and (C-A-B-D). The number of the remaining candidates for searching the best die partition case with the minimized number of TSVs becomes six, which equals these six candidates subtracted from the total candidates. To further reduce the candidates, the candidates in which neighboring partitions C and D have the second fewest connections between them are subtracted from the remaining six candidates, and the duplicated candidates in which partitions A, C, and D are neighbors at the same time are compensated. Two candidates including (A-D-B-C) and (C-B-A-D) eventually remain, and we can determine the best die partition with the minimized number of TSVs by calculating the number of TSVs needed to connect all dies in these two remaining candidates. Among the twelve total candidates for a die partition with four partitions, the final candidate with the minimum number of TSVs will be one of two remaining final candidates after the aforementioned process of filtering has been applied.

Generalizing such an approach, there are $n!/2$ total candidates for a die partition with n partitions. The number of subtracted candidates involving two neighboring partitions with the fewest connections is calculated as follows:

$$2\cdot\frac{(n-1)!}{2}. \qquad (3)$$

When repeating the process that subtracts candidates that involve the two neighboring partitions with the fewest connections and compensates the duplicated candidates, the number of die partition candidates with the minimum number of TSVs is summarized as follows:

$$\#\text{of candidates} = \frac{(n-1)!(n-2i)}{2} + x > 0, \qquad (4)$$

where n denotes the number of partitions, i denotes the number of calculations that subtract candidates involving two neighboring partitions with the fewest connections to reduce the number of candidates, and x represents the duplicated candidates when reducing the candidates. In (4), x varies depending on several cases, including those in which one partition is duplicated, two partitions are duplicated, and three partitions are duplicated; therefore, x cannot be formulated in an equation. Instead, when repeating the subtracting process n times, considering the worst case that the duplicated partitions are as few as possible, the tendency in the number of candidates is shown in Fig. 3. Figure 3 shows that the number of candidates decreases as the number of calculations for eliminating the candidates, i, increases, until only one candidate remains as the last among the many candidates after the calculations. In the relationship between the number of partitions and the number of calculations for eliminations

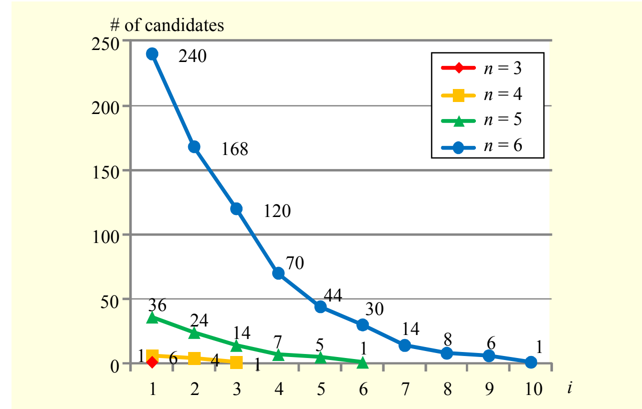

Fig. 3. Tendency in the number of candidates.

required in the search for a final candidate, i can be expressed as follows:

$$i = \sum_{k=3}^{n}(k-2) = \frac{(n-2)(n-1)}{2}. \qquad (5)$$

It is ultimately demonstrated that when there are n partitions, the calculations to eliminate the candidates are processed n^2 times and the final candidate for a die partition with the minimum number of TSVs can be searched.

IV. Thermal-Aware Floorplanning

When hot blocks gather in a specific spot on a die, the peak on-chip temperature increases because the power distribution on the die becomes uneven. Also, when hot blocks are placed in vertically neighboring spots, peak on-chip temperature increases because the power density on several dies becomes larger. The proposed thermal-aware floorplanning algorithm places hot and cool blocks alternately on each die for even power distribution. In the floorplanning stage, the block placement generally starts from the bottom-left of the die. In this case, the floorplanning tends to result in the formation of a fan shape in the top-right of the die compared with a dense and packed placement in the bottom-left of the die; therefore, there is no guarantee of an even power distribution throughout the die. To supplement this weak point, a die is divided into several quadrangles (bins) whereby the hot and cool blocks are placed alternately within each bin. By placing the blocks from the bottom-left of each bin for the dies with odd numbers and from the top-right of each bin for the dies with even numbers, the power density between neighboring dies is prevented from rapidly increasing. The number of bins is dependent upon the number of blocks and is defined by the following:

$$2 \le \sqrt{\#\text{bins}} \le \left\lceil \sqrt{\frac{\#\text{blocks}}{\alpha}} \right\rceil, \qquad (6)$$

http://dx.doi.org/10.4218/etrij.14.0113.1204

where α denotes the maximum number of blocks in a bin and [x] denotes the largest integer not greater than x as the maximal integer function.

When proceeding with the floorplanning stage by placing hot and cool blocks alternately, it is necessary to select the optimal bin in which to place a block so as to avoid high power density near the boundary line of the bins, as well as to make a more even power distribution throughout the die. The optimal bin (in terms of the power distribution) is determined by the distances between a block and its adjacent blocks and the power consumption of the blocks. Smaller distances between two blocks correspond to greater effects on the temperature. Likewise, higher power consumptions of two adjacent blocks correspond to greater effects on the temperature; therefore, the block should be located in the bin that minimizes the following equation:

$$\sum_{j=0}^{n-1} \frac{P_i P_j}{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2 + \left(z_i - z_j\right)^2}, \qquad (7)$$

where P_i and P_j denote the power consumption of block i and the jth block adjacent to block i, respectively. The x, y, and z coordinates for the center of block i are denoted by x_i, y_i, and z_i, respectively. And x_j, y_j, and z_j denote the x, y, and z coordinates for the center of block j, respectively. Every block has a large or small effect on temperature. Considering every block to calculate (7) considerably increases the computational complexity; therefore, it is necessary to calculate (7) with only adjacent blocks that have significant impacts on the temperature of the block directly. The conditions for adjacent blocks that directly affect the temperature of the block are as follows:

***z*-direction**: when projecting all dies as a plane, the blocks overlapped with block i influence the temperature of block i considerably.

***x*- and *y*-direction**: when expanding the quadrangular boundary of block i and drawing a circle with a radius equal to that of the block with the largest radius among the blocks that reach the extended boundary of block i, the blocks inside of the circle with the largest radius influence the temperature of block i considerably.

V. Simulation Results

In the simulation results, we compared the two proposed methodologies, respectively, with other conventional algorithms. Section V-*A* presents the comparison with the proposed min-cut die partition methodology and the conventional multiway partitioning algorithms. Section V-*B* presents the comparison with the proposed thermal-aware floorplanning methodology and the conventional floorplanning algorithms. Our proposed methodologies were implemented in the C++/STL programming language. All simulations were performed on a 64 bit Ubuntu Linux PC with 3 GHz CPU and 4 GB memory. MCNC benchmarks and GSRC benchmarks [13] were used for the simulations. Four dies were used for all circuits. The outside temperature is isothermal of constant room temperature, 27°C.

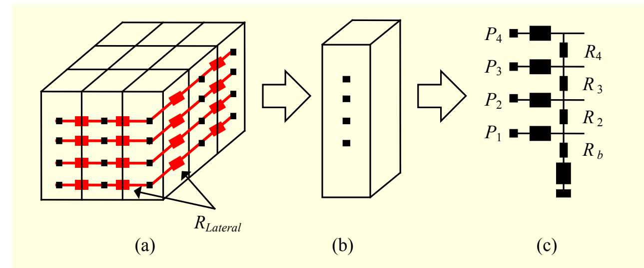

Fig. 4. Resistive thermal model for 3D IC.

To estimate temperature, we used the same thermal resistive model as in [10]. The 3D circuit is divided by a 2D array of tile stacks, as shown in Fig. 4(a); a tile stack is modeled as a resistive network. As shown in Fig. 4(b), each single tile stack is composed of several vertically stacked tiles, each of which connects to its neighbor at the same die by lateral thermal resistances, $R_{Lateral}$. Within each tile stack, a thermal resistor R_i is modeled for the ith die, while the thermal resistance of the bottom die material is modeled as R_b, as shown in Fig. 4(c).

The isothermal bases of room temperature are modeled as a voltage source. A current source is presented at every node in the network to represent the heat sources. One can spatially discretize the system and solve the following equation to determine the steady state thermal profile as a function of the power profile:

$$T = P A^{-1}, \qquad (8)$$

Table 1. Comparison for 3D die partition results.

Bench mark	[7]		[8]		[9]		Proposed	
	#TSVs	Run-time (sec)	#TSVs	Run-time (sec)	#TSVs	Run-time (sec)	#TSVs	Run-time (sec)
N10	132	0.37	113	0.33	147	0.15	108	0.21
N30	539	2.31	481	1.08	583	20.52	440	2.07
N50	726	45.61	778	21.32	819	2839.17	709	16.11
N100	1333	132.04	1767	15.80	1268	4581.45	1183	87.30
N200	3011	4503.78	3123	2019.33	2857	6711.83	2691	2501.69
N300	3711	8170.49	3574	4090.65	3346	10392.77	3218	4312.54
Avg. ratio	1.13	1.86	1.18	0.89	1.08	3.55	1.00	1.00

where A represents an $N \times N$ sparse thermal conductivity matrix, and T and $P(t)$ represent $N \times 1$ temperature and power vectors, respectively; N is the number of thermal conduction edges.

A. Results for Min-cut Die Partition

We compared the proposed min-cut die partition methodology with conventional methodologies for die partitioning in terms of the number of TSVs and run-time. References [7] and [8] present the multiway partitioning algorithms for 2D ICs. Reference [9] presents integer linear programs for 3D IC partitioning problems. The average results for the number of TSVs and run-time of the proposed methodology, for several benchmarks, are converted to 1.00 as a standard, as shown in Table 1. In terms of run-time, the proposed methodology improves by 1.8 times and 3.5 times over those of [7] and [9], respectively, but increases somewhat in comparison with [8]. However, the proposed methodology has the minimum number of TSVs, which is the key result in comparison with the other conventional methodologies for several benchmarks.

B. Results for Thermal-Aware Floorplanning

We compared the proposed thermal-aware floorplanning methodology with the conventional methodologies in terms of chip area, wirelength, the number of TSVs, and peak on-chip temperature. Reference [10] presents a simulated annealing-based thermal-driven floorplanning algorithm for 3D ICs. Reference [12] presents a force-directed floorplanning algorithm for 3D ICs. Similarly, the average results of the chip area, wirelength, the number of TSVs, and the peak on-chip temperature of the proposed methodology, for several benchmarks, are converted to 1.00 as a standard, as shown in Table 2. In terms of chip area and wirelength, the proposed methodology somewhat improves them. Moreover, the

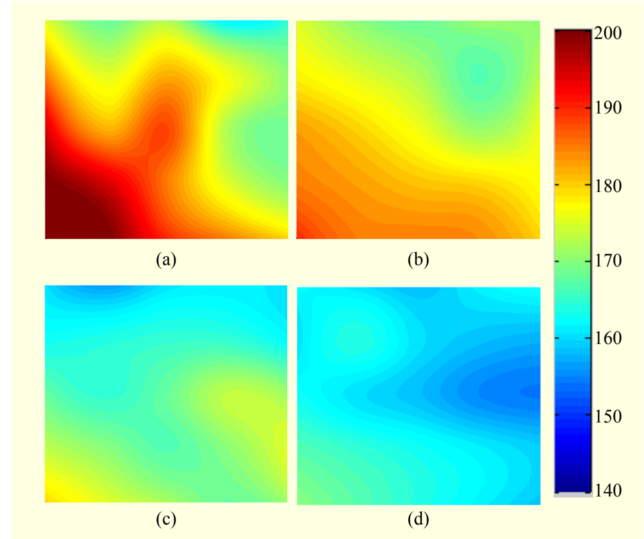

Fig. 5. Thermal maps of [12] for the ami33 circuit: (a) Die 4, (b) Die 3, (c) Die 2, and (d) Die 1.

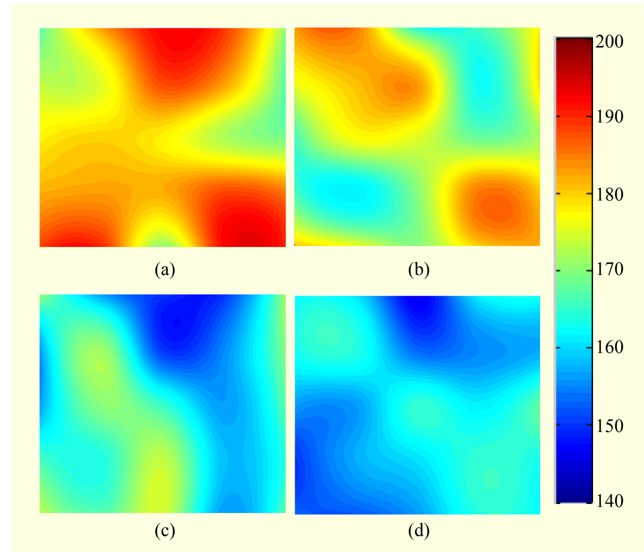

Fig. 6. Thermal maps of the proposed methodology for the ami33 circuit: (a) Die 4, (b) Die 3, (c) Die 2, and (d) Die 1.

Table 2. Comparison of thermal-aware floorplanning results.

Benchmark	[10]				[12]				Proposed			
	Area (mm^2)	HPWL (mm)	# TSVs	Temp (°C)	Area (mm^2)	HPWL (mm)	# TSVs	Temp (°C)	Area (mm^2)	HPWL (mm)	# TSVs	Temp (°C)
Ami33	43.3	24.4	121	212.4	40.1	23.9	118	201.3	43.1	25.1	108	192.4
Ami49	1,672.1	507.3	252	225.4	1,612.5	481.9	205	230.2	1,589.4	460.8	182	220.8
N100	6.6	122.1	949	172.4	6.6	92.6	749	156.8	6.7	93.9	712	142.1
N200	6.6	194.3	1,819	174.5	6.4	167.1	1,371	161.3	6.7	171.2	1,248	147.4
N300	10.5	306.1	2,267	190.7	9.8	239.8	2,056	167.1	9.5	241.6	2,001	154.2
Avg. ratio	1.05	1.16	1.27	1.14	1.01	1.01	1.06	1.07	1.00	1.00	1.00	1.00

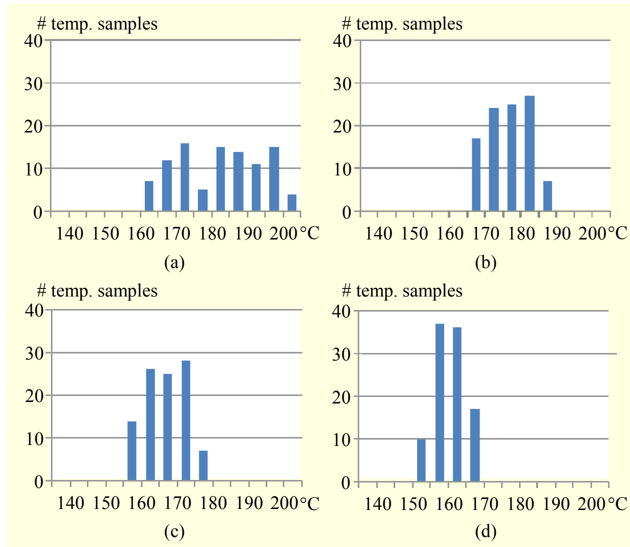

Fig. 7. Thermal histograms of [12] for the ami33 circuit.

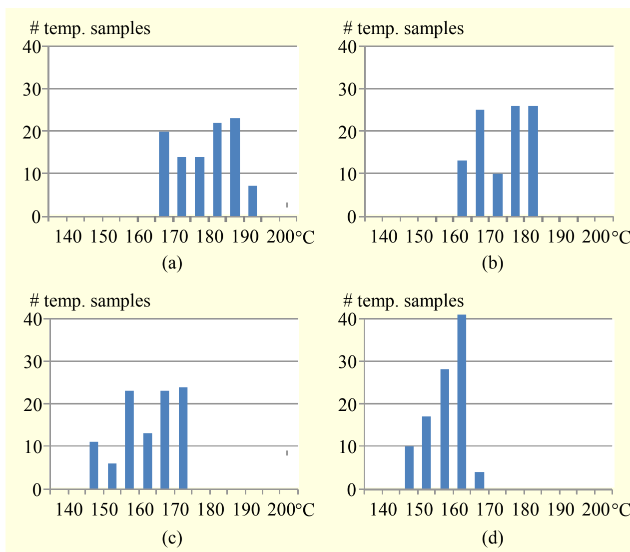

Fig. 8. Thermal histograms of proposed methodology for ami33 circuit.

proposed methodology has the minimum number of TSVs and a reduced peak on-chip temperature, which are the key results in comparison with the other conventional methodologies for several benchmarks.

Figures 5 and 6 show thermal maps of [12] and the proposed methodology for the ami33 circuit, respectively. In the thermal maps, the dark red area indicates a high temperature of around 200°C and the dark blue area indicates a low temperature of around 140°C. As shown in Figs. 5 and 6, the proposed methodology has a lower peak on-chip temperature than that in [12], and the overall temperature is distributed evenly across the entirety of the dies while minimizing the number of TSVs. Figures 7 and 8 show the thermal histogram results for Figs. 5 and 6, respectively. The *x*-axis of the histogram represents the temperature, and the *y*-axis of the histogram represents the number of temperature samples in the thermal map. As shown in Figs. 7 and 8, the proposed methodology has lower peak on-chip temperature than [12] in each die and has lower temperature distributions than [12].

VI. Conclusion

This paper proposes thermal-aware floorplanning methodology with min-cut die partition for 3D ICs. In the proposed min-cut die partition methodology, the number of TSVs is minimized based on the min-cut theorem considering a complementary set from the set of connections between two partitions when assigning the partitions to dies. In the proposed thermal-aware floorplanning methodology, the hot and cool blocks are placed alternately considering the distances between the blocks and the power consumption of the blocks to reduce the peak on-chip temperature and even out the power distribution in the dies. Compared to the conventional partitioning and floorplanning works for 3D ICs, the proposed methodologies reduced the number of TSVs and the peak on-chip temperature effectively while reducing the run-time.

References

[1] K. Banerjee et al., "3-D ICs: A Novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration," *Proc. IEEE*, vol. 89, no. 5, May 2001, pp. 602–633.

[2] C.-M. Hung and Y.-L. Lin, "Three-Dimensional Integrated Circuits Implementation of Multiple Applications Emphasizing Manufacture Reuse," *Comput. Digit. Techn., IET*, vol. 5, no. 3, May 2011, pp. 179–185.

[3] A. Todri et al., "A Study of Tapered 3-D TSVs for Power and Thermal Integrity," *IEEE Trans. VLSI*, vol. 21, no. 2, Feb. 2013, pp. 306–319.

[4] K. Puttaswamy and G.H. Loh, "Thermal Analysis of a 3D Die-Stacked High-Performance Microprocessor," *ACM GLS VLSI*, Apr. 2006, pp. 19–24.

[5] A. Jain et al., "Thermal Modeling and Design of 3D Integrated Circuits," *ITHERM*, Orlando, FL, USA, May 28–31, 2008, pp. 1139–1145.

[6] P. Li et al., "Efficient Full-Chip Thermal Modeling and Analysis," *ICCAD*, San Jose, CA, USA, Nov. 7–11, 2004, pp. 319–326.

[7] G. Karypis et al., "Multilevel Hypergraph Partitioning: Application in VLSI Domain," *IEEE Trans. VLSI*, vol. 7, no. 1, Mar. 1999, pp. 69–79.

[8] G. Karypis and V. Kumar, "Multilevel K-Way Hypergraph Partitioning," *ACM/IEEE DAC*, June 21–25, 1999, pp. 343–348.

[9] I.H.-R. Jiang and T.-W. Mei, "Generic Integer Linear Programming Formulation for 3D IC Partitioning," *SOCC*, Belfast, UK, Sept. 9–11, 2009, pp. 321–324.
[10] J. Cong, J. Wei, and Y. Zhang, "A Thermal-Driven Floorplanning Algorithm for 3D ICs," *ICCAD*, San Jose, CA, USA, Nov. 7–11, 2004, pp. 306–313.
[11] P. Zhou et al., "3D-STAF: Scalable Temperature and Leakage Aware Floorplanning for Three-Dimensional Integrated Circuits," *ICCAD*, San Jose, CA, USA, Nov. 4–8, 2007, pp. 590–597.
[12] Y. Huang et al., "A Thermal-Driven Force-Directed Floorplanning Algorithm for 3D ICs," *CAD/Graphics*, Huangshan, China, Aug. 19–21, 2009, pp. 497–502.
[13] P.H. Madden et al., *Standard Cell Benchmark Circuits*, Binghamton Laboratory for Algorithms, Circuits, and Computer Aided Design. Accessed Nov. 15, 2013. http://vlsicad.cs.inghamton.edu/benchmarks.html

Cheoljon Jang received his BS and MS degrees in electronics and computer engineering from Hanyang University, Seoul, Rep. of Korea, in 2011 and 2013, respectively. He is currently pursuing his PhD degree in nanoscale semiconductor engineering at Hanyang University. His current research interests include SoC design methodology, including physical design automation of 3D ICs.

Jong-wha Chong received his BS and MS degrees in electronics engineering from Hanyang University, Seoul, Rep. of Korea, in 1975 and 1979, respectively and a PhD degree in electronics and communication engineering from Waseda University, Tokyo, Japan, in 1981. Since 1981, he has been a professor of the Department of Electronics Engineering at Hanyang University. From 1979 to 1980, he was a researcher at the C&C Research Center of Nippon Electronic Company, Tokyo, Japan. From 1983 to 1984, he was a visiting researcher at the Korean Institute of Electronics & Technology, Seongnam, Rep. of Korea. In 1986 and 2008, he was a visiting professor at the University of California, Berkeley, USA. He was the chairman of the CAD & VLSI society of the Institute of the Electronic Engineers of Korea in 1993 and the president of the IEEK and of the KIEEE in 2007 and from 2009 to 2010, respectively. He is currently the Chairman of the Fusion SoC Forum. His current research interests are SoC design methodology (including memory centric design and physical design automation of 3D ICs); indoor wireless communication SoC design for ranging and location; video systems; and power IT systems.

http://dx.doi.org/10.4218/etrij.14.0113.1204