

An Energy and Traffic Aware Mapping Method for 3-D NoC-Bus Hybrid Architecture

Taotao Zhang, Ning Wu, Fang Zhou, and Lei Zhou

Abstract—3-D integrated circuits (ICs) increase the device density and reduce the delay of interconnects of dies by stacking two or more dies vertically. To take advantage of less energy consumption in the vertical direction and to solve the defect of poor heat dissipation, a mapping method for 3-D NoC-Bus hybrid architecture is proposed based on the task graph. To achieve the two optimal goals, optimal energy consumption and optimal balanced distribution of traffic, we first calculate the optimal combinations of buses using backtracking algorithm and then filter out the optimal order of the nodes on the buses by genetic algorithm. Experimental results show that the optimal energy consumption can reduce transmission energy by 35.5% compared with the optimal balanced distribution of traffic, and the latter's variance of traffic distribution among stacks and horizontal layers is only 5% of the former.

Index Terms—network-on-chip (NoC), 3-D NoC-Bus, task graph, interconnect energy, balanced distribution of traffic

I. INTRODUCTION

IN 3-D integrated circuits (ICs), by using through silicon vias (TSVs), multiple planar device layers are stacked and bond together, as shown in Fig. 1. As the gap between each layer is only tens of micrometers, the length of TSV is much smaller than that of the horizontal link, which is only one percent of the latter. As the interconnect distance between vertical adjacent cores is very small, little resistance and capacitance is caused in TSVs. So when transmitting the same amount of data, TSVs consume much less energy than horizontal links [1].

To make better use of the advantages of TSV in 3-D ICs, the vertical interconnect utilizes a wide, low-latency multi-drop bus, which forming 3-D NoC (network-on-chip)-Bus hybrid architecture [2]. The cores in the same stack are controlled by a centralized Arbiter, in which flit can be accessed in only one hop.

Manuscript received July 02, 2014; revised August 05, 2014. This work was supported by the National Natural Science Foundation of China under Grant (No.61376025, No.61106018), the Industry-academic Joint Technological Innovations Fund Project of Jiangsu under Grant (No.BY2013003-11), the Funding of Jiangsu Innovation Program for Graduate Education under Grant (No.KYLX_0273), and the Fundamental Research Funds for the Central Universities.

Taotao Zhang, the College of Electrical and Information Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), Nanjing 210016, China (email: taotzhang@163.com).

Ning Wu, the College of Electrical and Information Engineering, NUAA, Nanjing 210016, China (email: wunee@nuaa.edu.cn).

Fang Zhou, the College of Electrical and Information Engineering, NUAA, Nanjing 210016, China (email: zfnuaa@nuaa.edu.cn).

Lei Zhou, the College of Information Engineering, Yangzhou University, Yangzhou 225009, China (email: tomcat800607@126.com).

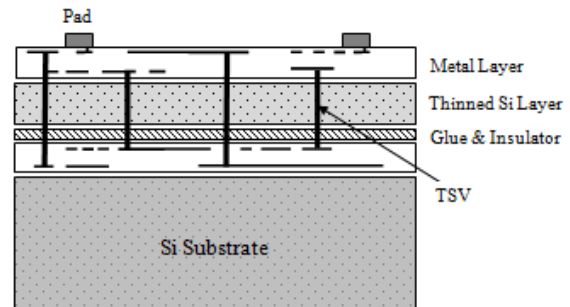


Fig. 1. 3-D integration using TSVs

However, 3-D ICs greatly increased device density, which introduces serious thermal issues. As high temperature can cause the chip unstable and reduce the chip's lifetime, thermal issue becomes a major challenge for 3-D integration. In the literature, many methods have been proposed, such as incorporating thermal vias into integrated circuits to mitigate thermal issues by lowering the thermal resistance of the chip [3], planning thermal via locations during floorplanning stages [4], cooling each level in the 3-D stack by microfluidic interconnects [5], scheduling dynamic thermally-aware job for 3-D systems [6], [7], designing an OS-level scheduling algorithm that performs thermal-aware task scheduling on a 3-D chip [8], etc.

In this paper, through distributing traffic uniformly to make an even heat distribution, a new method for mapping the task graph to 3-D NoC-Bus hybrid architecture is proposed. In addition to solve thermal issues, this algorithm can also reduce communication energy by restricting tasks within the same stack.

II. PRINCIPLES OF THE PROPOSED METHOD

Unlike the traditional 3-D NoC architecture, 3-D NoC-Bus hybrid architecture uses bus communication in the vertical direction, as shown in Fig. 2.

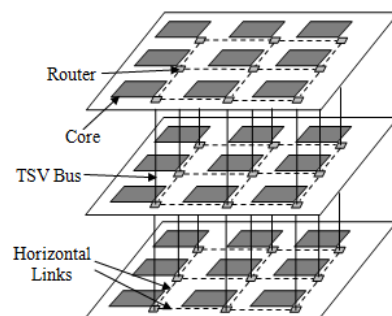


Fig. 2. 3-D NoC-Bus hybrid architecture

Since the architecture only needs one hop communicating on the bus, it greatly improves latency performance of system in the 3-D NoC-Bus hybrid architecture.

Because of the difference of transmission performance between TSV and the horizontal link, it takes two steps to map the task graph to the 3-D NoC-Bus hybrid architecture.

First, map the nodes of the task graph to the stacks (buses) which need to break the tree structure of the task graph into links using backtracking algorithm. It starts with the root node of the tree and goes to the child node of current node. If current node that is a cross point has multiple nodes, go small node first. Until reach the end of the link, go back to the cross point to find another link. Traversing all the links in the steps of the number of layers of 3-D NoC-Bus hybrid architecture, which is also the number of nodes on the bus, and the bus units can be found. Combining those units to make combinations of buses without a same node and screening out the combinations that meet the requirements.

Second, map the nodes on the buses to different layers. Because of a permutation and combination of nodes on the buses is a mapping result, there will be many results if there are many nodes on a bus. We use genetic algorithm to generate new results and filter the bad results generation to generation. Select the result meet the needs best in the last generation as the last mapping result.

III. TASK MAPPING

A. Task Graph Analysis

In order to fully describe the mapping algorithm in this paper, we use Task Graphs for Free (TGFF) [9] to generate a 20-node task graph randomly, as shown in Fig. 3.

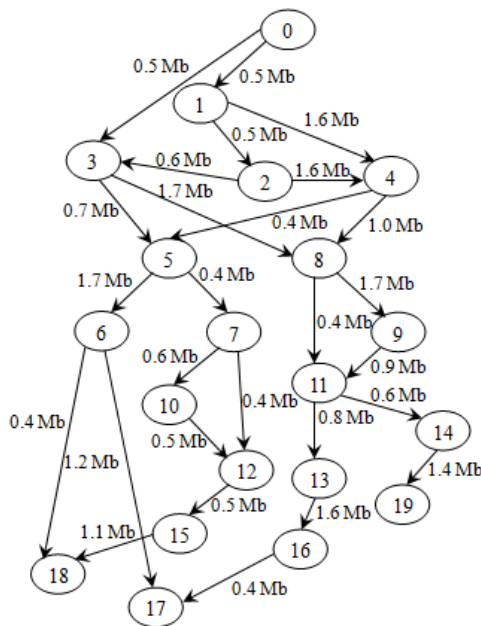


Fig. 3. 20-node task graph

Task graph is an acyclic directed graph derived from a special foreseeable application. In the graph, every node represents a task to be assigned to the core. Nodes connect to each other if they have communication requirement with other

nodes. The weight on the edge denotes the communication data volume between the corresponding nodes.

According to the number N of nodes in the task graph and the layers L need to be generated, an $n \times n \times L$ 3-D NoC-Bus hybrid architecture can be finalized via the following inequation and equation:

$$\begin{cases} (n-1)^2 < N_L \leq n^2 \\ N_L \geq N/L \end{cases} \quad (1)$$

N_L is the minimum integer bigger than N/L , which denotes the maximum nodes in the task graph can be allocated in a layer. If $N_L < n^2$, other $n \times n \times L - N$ nodes should be added to the $n \times n \times L$ 3-D NoC-Bus hybrid architecture.

When the task graph in Fig. 3 is mapped to the $3 \times 3 \times 3$ 3-D NoC-Bus hybrid architecture, 2 layers of the architecture will be allocated 7 nodes, and 6 nodes in the remaining layer. In addition, it takes 7 additional nodes to fill the $3 \times 3 \times 3$ architecture. Since the communication between the nodes are very different, how to layout this nodes will make great impact on system performance. Considering transmitting in TSVs not only can reduce much energy than in horizontal links [1] but also has a greater influence on the heat dissipation [10], [11], [12], the promised algorithm maps the nodes in the stack first and then maps the nodes in the horizontal layers.

B. Mapped to the Stacks

To assign the nodes in the task graph to the stacks, backtracking algorithm is utilized. The proposed method break up the tree structure into several links according to the parent-child relationship nodes.

First, look for links from the root node and go the path of small node preferentially when come across a bifurcate point. Until reach the end node, then return to the bifurcate point and find other links recursively.

Doing in this way can ensure that the correlative nodes are allocated in the same stack. As shown in Fig. 3, to map the task graph in a $3 \times 3 \times 3$ 3-D NoC-Bus architecture, 20 nodes should be mapped to 7 stacks, one of which contains two

TABLE I
LINKS FOUND FROM THE TASK GRAPH

Link	No.
0-1-2-3-5-6-17	1
0-1-2-3-5-6-18	2
0-1-2-3-5-7-10-12-15-18	3
0-1-2-3-5-7-12-15-18	4
0-1-2-3-8-9-11-13-16-17	5
...	...
0-3-8-9-11-14-19	30
0-3-8-11-13-16-17	31
0-3-8-11-14-19	32

nodes and each of the other contains 3 nodes. Done in accordance with the above, the task graph can be broken into 32 links, as shown in Table I.

Second, after finding out all the links, traverse the nodes on the links using L as a unit. Put the units that don't contain the

same node together as a combination of stacks. If the number of units that don't contain the same node is smaller than the number of stacks we need, find other units using $L - 1$ as a unit in the rest nodes, which are not in the units, as shown in Fig. 4.

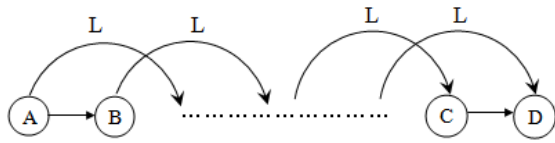


Fig. 4. Method of finding the L units in a link

A, B, C and D represent the nodes in a link of the task graph. L is the number of layers (the number of nodes a stack includes).

As is calculated, bit transmission cost the horizontal link 0.127 pJ and the TSV 9.56×10^{-3} pJ [1]. Bit transmission energy for TSV is only 7.5% of that of horizontal link. So it can reduce plenty of communication energy if aggregates tasks within the same stack. But the more tasks cluster in the same stack, the more power density of this stack increase, which will be likely to generate a hot spot.

This paper maps the nodes of the task graph in two extreme ways, the optimal energy consumption and the optimal balanced distribution of traffic. The former only considers aggregate as many tasks as we can in the same stack to save maximum communication energy and the latter just takes an average allocation in the stacks to make a uniform distribution of heat. According to the needs, considering them both is a good method to make a tradeoff between energy consumption and heat dissipation. They can be gotten at the same time due to they are not an inverse relationship.

1) Optimal Energy Consumption

6 stacks of 3 nodes can be found from the links in Table I, and the rest two nodes is allocated in the last stack. As the proposed method computed, there are 241 kinds of combinations. When only considering energy consumption, aggregating tasks in the same stack, using F_{stacks} as the degree of aggregating tasks, result can be reached when F_{stacks} is the maximum of the combinations as shown in Fig. 5. The "*" in Fig. 5 represents the node additional added.

$$F_{stacks} = \sum_{i=1}^n F_{vi} \quad (2)$$

F_{vi} represents the communication data volume of each stack.

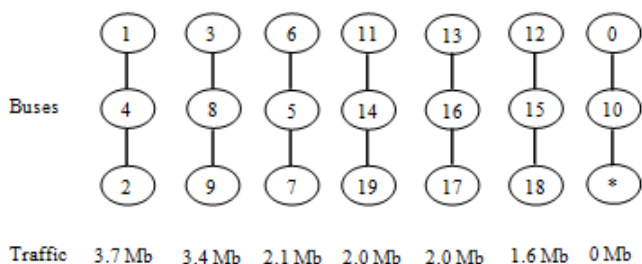


Fig. 5. Map of stacks of optimal energy consumption

2) Optimal Balanced Distribution of Traffic

To allocate the tasks averagely to the 7 stacks, which can obviously form a uniform distribution of heat to solve the thermal issues, we use S_v , the variance among the communication data volume of the stacks, as a yardstick to measure the degree of uniformity of traffic in each stack.

$$S_v = \sum_{i=1}^n (F_{vi} - E)^2 \quad (3)$$

E denotes the average value of all the n stacks. Via this equation, the result of optimal balanced distribution of traffic can be found when S_v is the minimum of the combinations, as shown in Fig. 6.

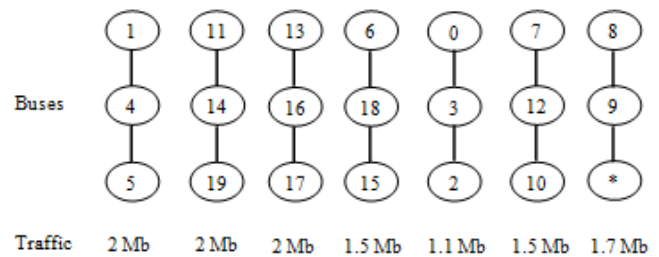


Fig. 6. Map of stacks of optimal balanced distribution of traffic

C. Mapped to the Horizontal Layers

Except the two optimal combinations of buses above, we can also find other combinations that meet our requirements and unite the result mapping to the horizontal layers to find the best mapping result among them.

Due to the transmission between nodes on the bus just need a single hop, it has little impact how to arrange the nodes on the stack. So the nodes on the bus can be sorted casually and what the proposed method need to do is finding out which arrangement of the nodes on the buses is the best. Genetic algorithm is used to find the best permutation we need. A model of chromosome is created in this algorithm, as shown in Fig. 7.

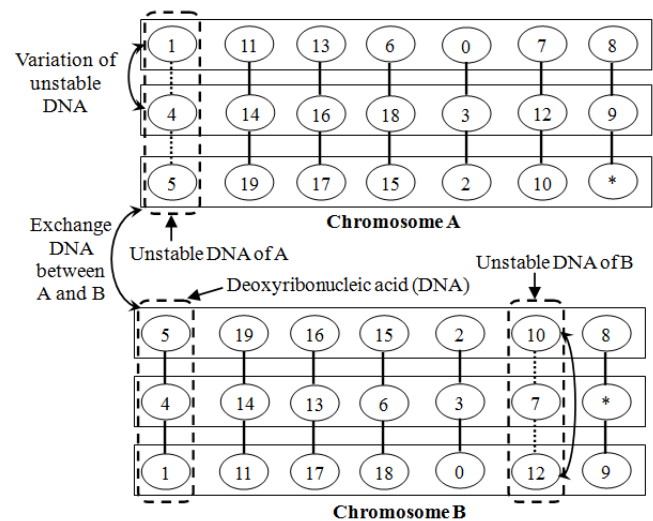


Fig. 7. The way of chromosomal heredity of the genetic algorithm of the proposed method

The length of the chromosomal which has multilayer structure is equal to the number of bus need to be mapped. A bus represents a deoxyribonucleic acid (DNA) in the chromosome. Every chromosome has only one unstable DNA (connected with dashed lines in Fig. 7) that is susceptible to mutation. A mutation or exchange of chromosomes is equivalent to change the nodes sort on the buses.

To choose the best chromosomes we need, a list of total communication data volume belong to each node is created. Encode each DNA in the chromosomes to sum the total communication data volume of each layer. To achieve a balanced traffic of each layer, we use S_L , variance among the communication volume of the layers, to measure the degree of uniformity of traffic in each layer.

$$\begin{cases} S_L = \sum_{i=1}^L \left(\sum_{j=(i-1) \times (n+1)}^{i \times n} F_j - E_L \right)^2 \\ E_L = \sum_{i=1}^{L \times n} F_i / L \times n \end{cases} \quad (4)$$

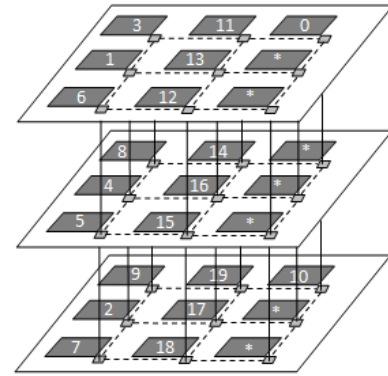
L represents the number of layers of chromosomes. E_L represents the average value of the layers. F_j denotes the total communication volume flowing through the node j . j is recoded according to the structure of chromosomes. For example, F_{10} of Chromosome A in Fig. 7 represents the total communication volume of node 16 of the task graph in Fig. 3, which is 2Mb.

Unlike the very low probability of gene mutation in the nature, here the mutation should have high probability to keep the diversity each generation. Then sort chromosomes in ascending order by S_L and keep the previous 50 chromosomes that meet the needs and reset all the unstable DNAs in the chose chromosomes to ensure diversity each generation. Until the last generation, choose the first chromosome as the best mapping result of horizontal layers.

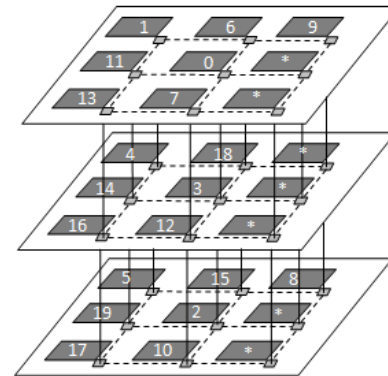
In the experiment, the initial population size is 100, the crossover rate is 0.8, the mutation rate that is big here is 0.5, the maximum evolution generation is 500 and the length of chromosome is 7. The mapping results of horizontal layers of optimal energy consumption and optimal balanced distribution of traffic, which respectively are $\{\{3, 1, 6, 11, 13, 12, 0\}; \{8, 4, 5, 14, 16, 15, *\}; \{9, 2, 7, 19, 17, 18, 10\}\}$ and $\{\{1, 11, 13, 6, 0, 7, 9\}; \{4, 14, 16, 18, 3, 12, *\}; \{5, 19, 17, 15, 2, 10, 8\}\}$, can be reached with the parameters being set above.

IV. MAPPING RESULTS ANALYSIS

In the experiment of this paper, we found two optimal mapping results, optimal energy consumption and optimal balanced distribution of traffic, based on the task graph in Fig. 3. Unite the mapping results of stacks and the mapping results of horizontal layers, the final mapping results of 3-D NoC-Bus hybrid architecture can be derived as shown in Fig. 8.



(a) Optimal Energy Consumption



(b) Optimal Balanced Distribution of Traffic

Fig. 8. Mapping results of the two optimal goals of 3-D NoC-Bus hybrid architecture

Assuming D is the Manhattan distance between two nodes in the 3-D NoC-Bus hybrid architecture.

$$D = D_{horizontal} + D_{vertical} \quad (5)$$

So the $D_{horizontal}$ of node 11 and node 8 of the optimal balanced distribution of traffic in Fig. 8 is 3 and the $D_{vertical}$ is 2. Using the Manhattan distance, we can calculate the energy consumption during communication by the following equation.

$$\begin{cases} E_{link} = E_{horizontal} \times D_{horizontal} + E_{vertical} \times D_{vertical} \\ E_{horizontal} = 1/2 \times \alpha \times C_{horizontal} \times V_{dd}^2 \end{cases} \quad (6)$$

$E_{horizontal}$ is determined by the physical capacitance of horizontal link $C_{horizontal}$, supply voltage V_{dd} and activity factor α of the network communication [1]. $E_{vertical}$ can be calculated in a similar manner like $E_{horizontal}$ in (6).

According to the technology mentioned in literature [1], bit transmission energy can be respectively calculated as 0.127 pJ and 9.56×10^{-3} pJ of horizontal link and vertical link. With that, the total transmission energy of optimal energy consumption is 2.05 uJ, just 64.5% of the transmission energy of optimal balanced distribution of traffic, which is 3.19 uJ.

Although optimal energy consumption can save much energy than optimal balanced distribution of traffic, but the traffic distribution has large gap between stacks and layers, which is not conducive to system heat dissipation. The difference of traffic distribution between optimal ways is

obvious, as shown in Fig. 9.

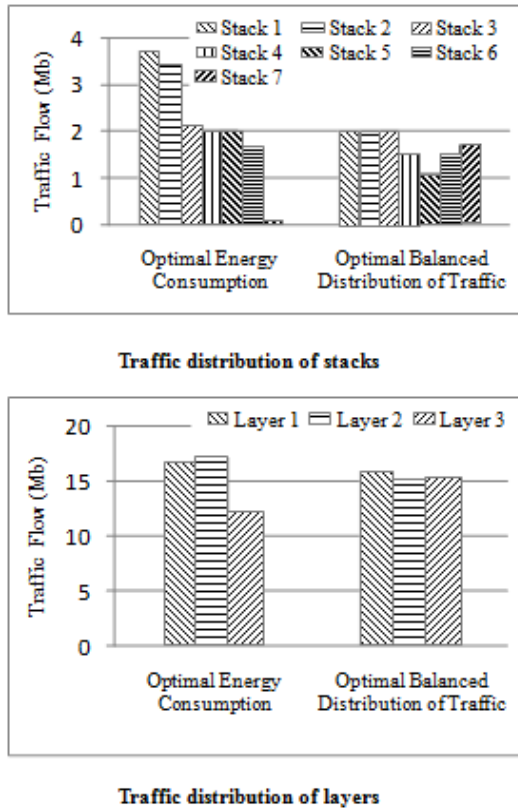


Fig. 9. Traffic distribution of the two optimal goals of 3-D NoC-Bus hybrid architecture

No matter among stacks or layers, it is obvious that the traffic distribution is more uniform in optimal balanced distribution of traffic than optimal energy consumption. The sum of variances of the stacks and layers' communication data volume of the optimal balanced distribution of traffic is only 5% of the optimal energy consumption.

The two optimal ways tested here are two extreme ways, one of which only considering reducing transmission energy, and the other is just for the sake of heat dissipation. Considering both of them by uniting F_{stacks} in (2) and S_v in (3), the unified equations can be represented by (7).

$$\begin{cases} F_{stacks} \geq F_{min} \\ Min(S_v) \end{cases} \quad or \quad \begin{cases} S_v \leq S_{max} \\ Max(F_{stacks}) \end{cases} \quad (7)$$

F_{min} and S_{max} are set according to the results of bus combinations. On the basis of bus combinations in the step of mapping to stacks, we set $F_{min} = 12$ Mb and the map of optimal balanced distribution of traffic can be gotten as shown in Fig. 10.

Using (6), the total transmission energy in Fig. 10 is 2.665 uJ, accounting for 83.6% and 130.2% of optimal balanced distribution of traffic and optimal energy consumption. And the sum of variances of the stacks and layers' communication data volume is between the variances of the two optimal goals, too.

By setting some other constraints, the proposed method can be used further. Such as, find the relationship between the stack's temperature and communication data volume. Define a threshold temperature of stacks, and we can find the most

energy saving solution by using the proposed method under the circumstance that each stack's temperature is less than the threshold temperature.

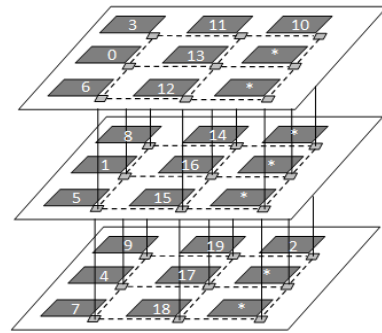


Fig. 10. Map of optimal balanced distribution of traffic under $F_{stacks} \geq 12$ Mb

V. CONCLUSIONS

The proposed method is used to find a good map to the 3-D NoC-Bus hybrid architecture based on the task graph. In the experiments, the optimal energy consumption can reduce transmission energy by 35.5% compared with the optimal balanced distribution of traffic, and the communication data volume of the equalization is more uniform than the optimal energy, which is conducive to heat dissipation. And the two optimal goals can be considered both by a compromise optimization goal between them.

REFERENCES

- [1] Cheng Y, Zhang L, Han Y, et al. Thermal-constrained task allocation for interconnect energy reduction in 3-d homogeneous mpsocs[J]. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, 2013, 21(2): 239-249.
- [2] Li, Feihui, et al. "Design and management of 3D chip multiprocessors using network-in-memory." ACM SIGARCH Computer Architecture News. Vol. 34. No. 2. IEEE Computer Society, 2006.
- [3] Goplen, Brent, and Sachin Sapatnekar. "Thermal via placement in 3D ICs." Proceedings of the 2005 international symposium on Physical design. ACM, 2005.
- [4] Wong, Eric, and Sung Kyu Lim. "3D floorplanning with thermal vias." Design, Automation and Test in Europe, 2006. DATE'06. Proceedings. Vol. 1. IEEE, 2006.
- [5] Bakir, Muhannad S., et al. "3D heterogeneous integrated systems: Liquid cooling, power delivery, and implementation." Custom Integrated Circuits Conference, 2008. CICC 2008. IEEE. IEEE, 2008.
- [6] Coskun, Ayse Kivilcim, et al. "Dynamic thermal management in 3D multicore architectures." Design, Automation & Test in Europe Conference & Exhibition, 2009. DATE'09.. IEEE, 2009.
- [7] Zhu, Changyun, et al. "Three-dimensional chip-multiprocessor run-time thermal management." Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on 27.8 (2008): 1479-1492.
- [8] Zhou, Xiuyi, et al. "Thermal-aware task scheduling for 3d multicore processors." Parallel and Distributed Systems, IEEE Transactions on 21.1 (2010): 60-71.
- [9] Dick, Robert P., David L. Rhodes, and Wayne Wolf. "TGFF: task graphs for free." Proceedings of the 6th international workshop on Hardware/software codesign. IEEE Computer Society, 1998.
- [10] Zhou, Xiuyi, et al. "Thermal-aware task scheduling for 3d multicore processors." Parallel and Distributed Systems, IEEE Transactions on 21.1 (2010): 60-71.
- [11] Banerjee, Kaustav, et al. "3-D ICs: A novel chip design for improving deep-submicrometer interconnect performance and systems-on-chip integration." Proceedings of the IEEE 89.5 (2001): 602-633.
- [12] Xie, Yuan, et al. "Design space exploration for 3D architectures." ACM Journal on Emerging Technologies in Computing Systems (JETC) 2.2 (2006): 65-103.