CC-002

TSV Placement for Large-size 3D-NoC by Evolutionary Algorithm

Yuting Huang[1]   Xin Jiang[2]   Takahiro Watanabe[3]
[1][3] Graduate School of Information, Production and Systems, WASEDA University
[2] National Institute of Technology, Kitakyushu College

Abstract

In recent years, Through Silicon Via (TSV) based Three-dimensional Network-on-Chip (3D NoC) is emerging as the most popular interconnection solution for the System-on- Chip (SoC) design. In this work, we propose a novel Evolutionary Algorithm (EA) for optimally allocating each TSV in the partially connected 3D NoC within the definite number of TSVs. The experimental results show that our proposed method not only achieves better performance, but also improves the stability of the network, compared with the method [1].

1. Introduction

As the scaling technology developed, now it allows us to build chips consisting of hundreds of millions of processor cores. However, efficient movement of data among large number of nodes is the mainly factor, which bring about Multi-Processor System-on-Chip systems becoming communication limited rapidly. [2] As an efficient and scalable communication solution, 3D-NoC fully explains the latest microelectronics techniques for increasing chip density while reducing power consumption.

For implement 3D integration, TSV is the most vertical stacking technique due to its low latency and low power consumption [2]. However, in designing the TSV based 3D NoC, there is a limitation on TSV number because the yield of chip drops sharply beyond a number threshold. In addition, the allocations of TSVs have a significant impact on the network performance. In this work, we design an optimal TSV placement method based on Evolutionary Algorithm within definite TSV number to achieve high performance and stability. This method applies direction based search to ensure searching efficiency and high quality of the solutions. In the experiments, compared with other TSV placement method, our proposed one achieves better network performance in most cases.

2. Related Work

There are several researches to improve the communication performance of the network while reduce costs. Some researchers have tried to provide low-cost and high-performance TSVs. [3] presented a post processing approach to minimize number of TSVs for 3D ICs. However, the performance metrics are not considered. Moreover, only small scale circuits are considered.

In [4], the authors presented a study on system performance of different number of TSV usage between layers and their configuration. They discussed the TSVs number required for a 3D NoC and different placements of layer connections by a 64-core 3D NoC. However, the method provided by this paper can only be used for symmetric mesh.

In [1], an Integer Linear Programming (ILP) based method for allocating and placing a minimal number of TSVs was proposed. However, in the second step of this algorithm, the placement is reconfigured according to the max redundancy so that the number of TSVs may be not the minimum number. Furthermore, uniform distribution can't be ensured.

3. Preliminary Theories

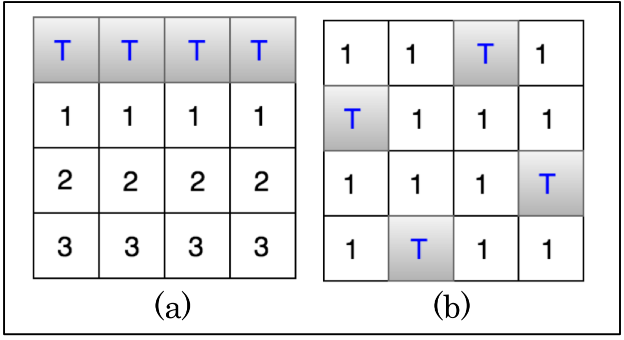
In designing the TSV based 3D NoC, it is not practical to use full connection in which every node is connected through a TSV, because the yield of chip drops sharply beyond a threshold number. In addition, the allocations of TSVs have a significant impact on the network performance. In this chapter, before designing the algorithm, we analyze the challenges in the TSV allocation problem, and formulate our design model. Then some preliminary knowledge will be described for designing the proposed algorithm.

3.1 Challenge in Design

To optimize the placement of TSV in the 3D NoC, network performance and reliability must be considered simultaneously. To improve network latency, uniform distribution for ensuring packets arrival within certain hops is a critical problem. Moreover, increasing the number of reachable neighbor nodes is a method to deal with transmission failure.

3.1.1 Uniform Distribution of TSV

Since the number of TSVs is limited, multiple network nodes have to share one TSV pillar for inter-layer communication. In the partially connected 3D network, an improper TSV distribution will increase the distance to reach the destinations, and further cause performance degradation in case of a high inter-layer communication. In the following example, two different TSV distributions are used in the 4*4*2 NoC with 4 TSVs (shown in Fig.1 (a) and (b)).



T	T	T	T
1	1	1	1
2	2	2	2
3	3	3	3

(a)

1	1	T	1
T	1	1	1
1	1	1	T
1	T	1	1

(b)

Fig. 1 TSV distributions on 4*4*2 mesh, Numbers mean hops from the nearest TSV

---

† 一般社団法人情報処理学会, IPSJ

‡ 一般社団法人電子情報通信学会, IEICE

We use Average Hop Count (AHC) to evaluate the transmission distance, which is defined as the accumulation of the hops of all nodes to transfer a flit to destinations divided by number of nodes.

In the first configuration (Fig.1(a)), all the TSV pillars are placed on one side of the chip, causing high delays on the other side of the chip (AHC = 24/16 = 1.5). When transfer data from source to destination, the path is: L1(0) > L1(4) > L1(8) > L1(12) > TSV > L2(12) > L2(8) > L2(4) > L2(0). Furthermore, due to the poor distribution of pillars, there could be traffic contention.

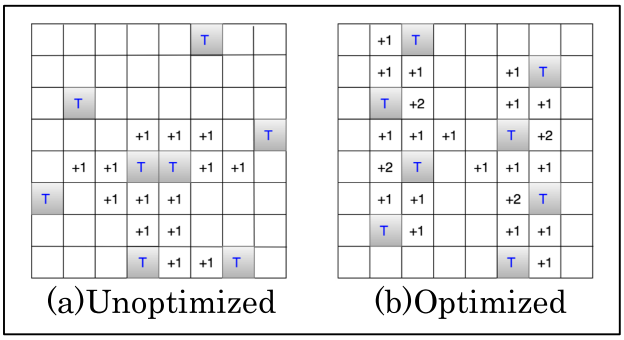
In the second configuration(Fig.1(b)), all nodes without a TSV pillar require just one hop to access a pillar, which results in a minimal average hop count (AHC = 8/16 = 0.5). When data is transferred from source to destination, the path is: L1(0) > L1(4) > TSV > L2(4) > L2(0). It is obvious that, the TSV pillars are well distributed.

Therefore, try to distribute the TSVs uniformly in the network is a good idea to balance the traffic and then improve the network latency.

3.1.2 Redundancy

In case of failure of a pillar, the communication between layers may be affected. To provide redundancy, each node shall be adjacent to at least one redundant TSV pillar.

This can be seen in the examples in Fig. 2 where two placements for an 8*8 mesh are shown. In these figures, a TSV pillar is reachable for a node within two hops. The number of a node indicates how many redundant pillars the node can reach, i.e. "+1" means one extra TSV pillar can be reached within two hops. Nodes without a number just can reach one pillar within two hops, and thus have no redundancy.



					T		
	T						
			+1	+1	+1		T
	+1	+1	T	T	+1	+1	
T		+1	+1	+1			
			+1	+1			
			T	+1	+1	T	

(a)Unoptimized

	+1	T					
	+1	+1			+1	T	
	T	+2			+1	+1	
	+1	+1	+1		T	+2	
	+2	T		+1	+1	+1	
	+1	+1			+2	T	
	T	+1			+1	+1	
					T	+1	

(b)Optimized

Fig. 2 Redundancy optimization

Despite the fact that both placements have the same minimal number of pillars, in the optimized placement (Fig. 2 (b)) more nodes can reach a redundant pillar than in the unoptimized placement (Fig. 2 (a)). While in the unoptimized placement only 14 nodes have one redundant pillar, in the optimized placement 18 nodes have one redundant pillar, and 4 nodes even have two redundant pillars.

Therefore, to tolerate the failure of a TSV pillar, each node shall reach as many TSVs as possible.

3.2 3D NoC Model

We use the same 3D NoC model in [1], in which there are two layers in the network, one layer for cores and the other for caches. The vertical links are implemented by TSVs.

4. Proposed algorithm of optimal TSV distributions

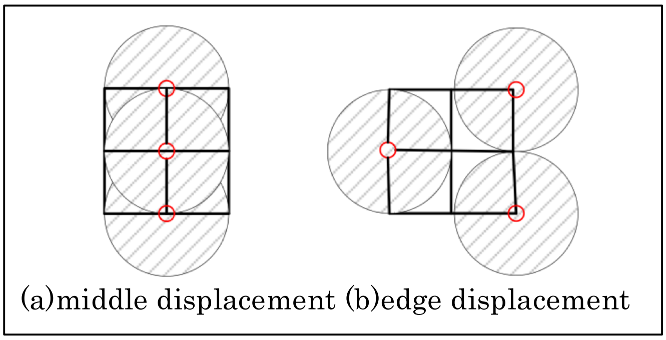
In this chapter, we give a detailed description on our proposed optimal TSV placement algorithm based on EA within definite TSV number to achieve high performance and stability. The routers are constructed as mesh topology, and the number of TSVs is given in advance. The problem is to search for an optimal TSV placement for ensuring stable message transmission and achieving better performance. In this work, we first design an objective function for distributing the TSVs uniformly and getting high redundancy which causes diversity in path selection. Then we design a direction based Evolutionary Algorithm to search for the best solution.

4.1 Objective Function

We define the maximal distance hmax as maximum distance in hops up to which a pillar $p_j \in P$ is considered to be reachable by a node $s_i \in S$. Also, the number of TSVs k_p is defined as the total number of pillars, which have to be placed in the network.

Given k_p and h_{max}, draw circles by using each TSV as the center and h_{max} as the radius. The circles should cover all the nodes (each node is reachable within the distance of h_{max}). The area covered by this circle is called effective area of the TSV. It is obvious that in the mesh the area is bigger the TSVs can effective more nodes. Then the objective can be represented as:

$$\text{Min } S = \left(\sum_{i=1}^{k_p} Area_{circle_i} - Area_i \cap Area_j\right) / Area_{mesh}$$



(a)middle displacement (b)edge displacement

Fig. 3 Example of the objective calculation

Fig. 3 shows an example of the objective function when hmax is 1, and k_p is 3 in the 3*3 mesh with 3 TSVs. In Fig. 3 (a), we assume 3 nodes in the middle region of mesh with TSVs. Also in Fig.3(b) we assume 3 nodes at the edge of mesh with TSVs. So the value of objective can be calculated as hatched area in Fig. 3 (a) and (b) by the mesh area. It is obvious that hatched area of Fig. 3(a) is smaller than that of (b). Therefore, according to the objective function, the condition Fig.3 (a) is better than (b).

4.2 Representation

We use a 2D coordinates to represent the positions of the TSVs. In the Optimal Placement of TSV problem, we use two bits (xy) to represent the coordinates of each TSV in the mesh.

---

† 一般社団法人情報処理学会, IPSJ
‡ 一般社団法人電子情報通信学会, IEICE

The chromosome is composed of a sequence of (xy) which represents the coordinate of each node having TSV. The length is equal to the number of TSVs.

4.3 Mutation

To search the solutions more efficiently and effectively, we apply direction based search to generate new offsprings. The basic theory is from Differential Evolution (DE) [5], which is one of the Evolutionary Algorithms.

Each chromosome individual is represented as a vector. The chromosome is represented as r $\{r_0, r_1, r_2, ..., r_n\}$. First, randomly select three difference individuals, r_i, r_m and r_n. Then, set r_i as the base point, its mutant r_i` is expressed as:

$$r_i` = r_i + (r_m - r_n) \quad \text{or} \quad r_i` = r_i + (r_n - r_m)$$

The feasible solution is the nodes in the first quadrant, which means the nodes within the mesh region. We select every individual as the base point in sequence, and randomly select the other two individuals to generate the offspring individual r` from r.

Let us see an example of mutation for one item (Fig. 4).

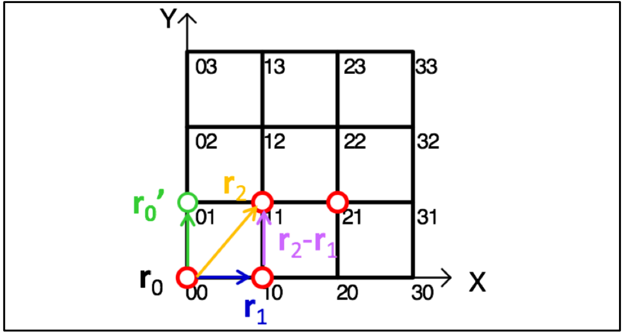


Fig. 4 Example of mutation for one item

Because $(r_1 - r_2)$ do not in the first quadrant, however, $(r_2 - r_1)$ in the first quadrant. We could generate that $r_0`=r_0 + (r_2 - r_1)$. The same, $r_1`$, $r_2`$, $r_3`$ could be generated by mutate parents. Hence, in the case we could get the chromosome $\{r_0`, r_1`, r_2`, r_3`\}$

The offspring is generated by the following rule:

$$r'_{ui} = \begin{cases} \mathbf{r}'_i \text{(if } r'_i \text{ is feasible, and } r'_i \text{ is different from other mutant individuals)} \\ \mathbf{r}_i \qquad \text{(if } r'_i \text{ is unfeasible, or } r'_i \text{ is the same with } r'_j\text{)} \end{cases}$$

4.4 Selection

The population should be decided before the program run. When replace population, we select 80% best solutions according to objective value, and the other 20% randomly select from the offspring. The program will be ended at the condition that achieve the maximum generation which had been set in advance.

5. Experiments of the Proposed Algorithm

In this section, we give the experimental result of optimized TSV placement generated by our proposed algorithm.

5.1 Experiment Setup

We build 3D NoC architecture as two layers mesh size from 3*3 to 16*16. The proposed method is compared with Integer Linear Programming proposed in [1].

The experiments are implemented by C# .net language on a 2 GHz Windows workstation. Also, we evaluate the performance of our TSV placement by using an open source simulator Noxim-3D [6]. We perform different experiments in order to evaluate our proposed method's performance in terms of latency and throughput. The parameters are shown in Table 1:

Table 1 Parameter

Name	Value
kp	Minimum value same as [1]
hmax	1,2
Simulation time	10000*20
Routing algorithm	Odd-even_Z, XYZ

5.2 Results for TSV Distribution

The TSV placement is executed for different sized NoC with hmax = 1 and hmax = 2, and some of the results are shown in Table 2:

Table 2 TSV distributions

Mesh Size	TSV Distributions		
	ILP ○	Propose ★ (h_{max}=1)	Propose △ (h_{max}=2)
3×3	1,5,6	1,4,7	2,4,6
4×4	1,7,8,14	1,7,8,14	5,6,9,10
5×5	2,5,9,12,15,19,22	2,5,9,12,15,19,22	6,7,8,12,16,17,18
6×6	0,3,7,11,14,18,22,27,31,35	2,5,6,15,16,19,29,30,32,33	7,9,10,14,19,20,22,25,27,29

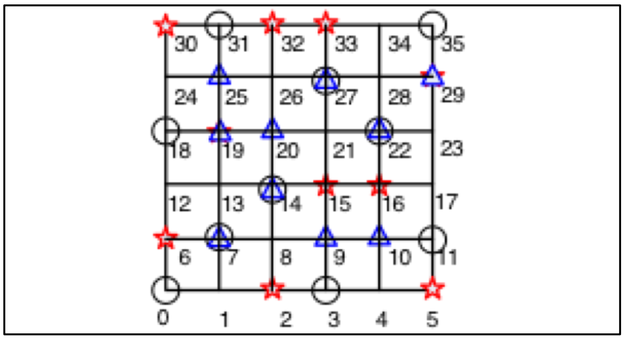


Fig. 5 TSV distributions of 6×6 mesh

Fig. 5 shows the example for 6*6 mesh size, it`s obvious that our method tries to distribute the TSVs in the middle region of the network. The reason is the objective function. The table 5.2 shows the distribution result of TSV from mesh size 3*3 to 16*16. As the result shows, when h_{max}=1 the results find by our proposed method some times are same with ILP based method. That is h_{max}=1 is the most rigorous constraint condition for partial connections, also the TSV numbers are the minimal need for the mesh size.

---

† 一般社団法人情報処理学会, IPSJ
‡ 一般社団法人電子情報通信学会, IEICE

5.3 Simulation results and analysis

These TSV pillars distribution results are input to the simulator Noxim. The performance including average latency, throughput and energy are evaluated. Routing algorithm including Odd-even Z routing algorithm and XYZ routing at random traffic are used. We analysis the results from 3*3*2 and 16*16*2. Fig. 6 show the result of 6*6*2 mesh with XYZ routing.

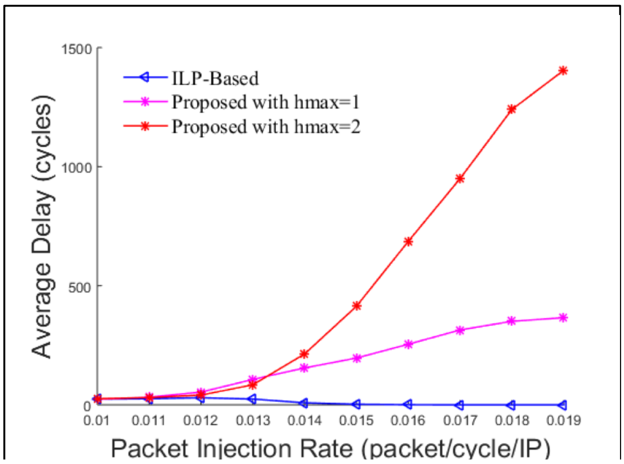


Fig. 6 (a) 6×6×2 with XYZ routing at random traffic – average latency

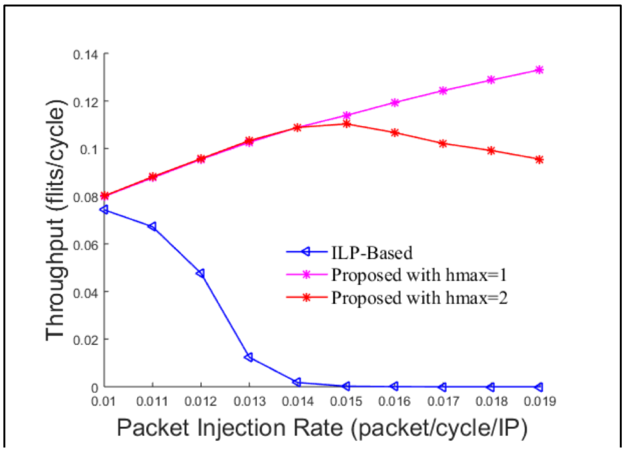


Fig. 6 (b) 6×6×2 with XYZ routing at random traffic – throughput

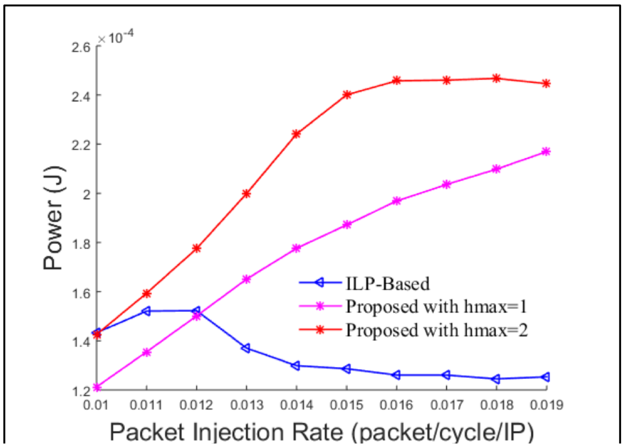


Fig. 6 (c) 6×6×2 with XYZ routing at random traffic – energy

As a result, the proposed algorithm achieves lower latency and energy and higher throughput compared with the ILP method.

In addition, in Fig 6 (b), there is an injection rate point from which the throughput drops sharply, that is most packets can not reach the destination with this TSVs distribution. It also means, the network becomes unstable for message transmission. Therefore, our proposed algorithm can sustain higher injection rate compared with the ILP method.

We also implement the experiment in the larger scale network, and the results are also that our proposed method can improve the performance and stability of the network.

6. Conclusion

We proposed an Evolutionary Algorithm based TSV placement optimization Algorithm that directed based search for efficient search, also ensure feasibility of the solutions. An objective function for effectively allocate TSVs on the network with limited TSV number was introduced. The experimental results show that out method improves the stability of the network.

Reference

[1] Xu TC et al., "Optimal placement of vertical connections in 3d network-on-chip," Journal of Systems Architecture 2013, pp. 441–454.

[2] William James Dally, Brian Towles, “Principles and Practices of Interconnection Networks” [M]. Printed in the United States of America: MORGAN KAUFMANN PUBLISHERS, 2003.

[3] C.-H. Lee et al., "A post-processing approach to minimize TSV number for high-level synthesis of 3D ICs," in: 2010 International Symposium on 3CA, vol. 1, 2010, pp. 434–437.

[4] T.C. Xu et al., “Optimal number and placement of Through Silicon Vias in 3D Network-on-Chip,” in: 2011 IEEE 14th DDECS, 2011, pp. 105–110.

[5] Campbell H, Dudek R, Smith M. A heuristic algorithm for the n job, m machine sequencing problem[J].Management Sxience, 1970, 16(10):630-637.

[6] Vincenzo Catania, Andrea Mineo, Salvatore Monteleone, Maurizio Palesi, and Davide Patti: " Cycle-Accurate Network on Chip Simulation with Noxim," ACM Trans. Model. Comput. Simul. 27, 1, pp.1-25, Article 4 (August 2016).