

A Sophisticated Routing Algorithm in 3D NoC with Fixed TSVs for Low Energy and Latency

XIN JIANG^{1,a)} LIAN ZENG^{1,b)} TAKAHIRO WATANABE^{1,c)}

Received: December 6, 2013, Revised: March 14, 2014,
Accepted: April 28, 2014, Released: August 4, 2014

Abstract: With rapid progress in Integrated Circuit technologies, Three-Dimensional Network-on-Chips (3DNoCs) have become a promising solution for achieving low latency and low power. Under the constraint of the TSV number used in 3DNoCs, designing a proper routing algorithm with fewer TSVs is a critical problem for network performance improvement. In this work, we design a novel fully adaptive routing algorithm in 3D NoC. The algorithm consists of two parts: one is a vertical node assignment in inter-layer routing, which is a TSV selection scheme in a limited quantity of TSVs in the NoC architecture, and the other is a 2D fully adaptive routing algorithm in intra-layer routing, which combines the optimization of routing distance, network traffic condition and diversity of the path selection. Simulation results show that our proposed routing algorithm can achieve lower latency and energy consumption compared with other traditional routing algorithms.

Keywords: 3D NoC, fully adaptive, routing algorithm

1. Introduction

Highly advanced technologies in Integrated Circuit (IC) design and increasing transistor counts on a chip have led to a wide use of Networks-on-chip (NoC) instead of conventional interconnects due to its features such as scalability, reliability and higher bandwidth. On the other hand, increasing the number of transistors on a chip has made Three-Dimensional Integrated Circuits (3D ICs) technology a promising solution for achieving low latency and low power. Three-Dimensional Networks-on-chip (3D NoCs) combine both benefits of short vertical interconnects of 3D ICs and the scalability of NoCs [13]. Wafer-to-wafer bonding technology, where the vertical interconnects are implemented using Through Silicon Vias (TSVs), is one of the popular choices for 3D integration [14]. In the 3D NoC interconnect network, vertical links are integrated in the 3D routers and can be connected by TSVs from one layer to another. However, in real 3D NoC design, the number of TSVs is limited to a certain value due to the yield of 3D IC. **Figure 1** shows the yield trend on TSV numbers in different processes [13]. For all processes, there is a clear upper bound after which chip yield drops sharply [15], and this upper bound is always considered as a constraint on designing 3D NoC architectures [21]. Another reason for reducing the number of TSVs is to save the area. According to Ref. [26], let's consider a 3D NoC with a hundred 64-bit vertical TSV links between layers. The TSVs will occupy a considerable area of a computation core. Consuming too much area on a chip does not only make

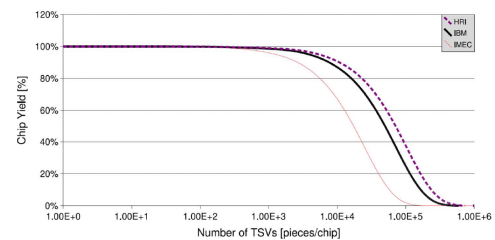


Fig. 1 Relation between yield and TSV number [13].

a challenge to floorplanning and routing, but also increase the overhead of the chip design. Therefore, designing a proper routing algorithm with fewer TSVs has become a challenging topic in the recent research on 3D NoC applications.

In this paper, we propose a fully adaptive routing algorithm in 3D mesh NoC topology under the constraint of limited quantity of TSVs which are distributed beforehand. The basic scheme of the routing is to firstly route the packet from the source node to the destination layer through proper vertical routers (we call it inter-layer routing), and then route to the destination node using a 2D routing algorithm within the destination layer (we call it intra-layer routing). The algorithm consists of two parts: one is a vertical node assignment in inter-layer routing, which is a TSV selection scheme under the constraint of a limited quantity of TSVs in NoC architecture, and the other is a 2D fully adaptive routing algorithm in intra-layer routing, which combines the optimization of routing distance, network traffic condition and diversity of the path selection. The inter-layer routing algorithm assigns a proper vertical router according to the source and destination nodes for the shortest path transmission, which outperforms the traditional static method. The feature of our proposed routing algorithm is that it takes distance, network state and path

¹ Graduate School of Information, Productions and Systems, Waseda University, Kitakyushu, Fukuoka 808–0135, Japan

a) jiangxin@ruri.waseda.jp

b) lian_zeng_ips@ruri.waseda.jp

c) watt@waseda.jp

diversity into consideration at the same time. It is different from other adaptive routing algorithms which choose a route among the shortest routes by using information about the network state in making the routing decision at each hop [1]. The application of inter-layer and intra-layer routing algorithms achieves high enhancement of network performance in 3D NoC. The simulation results demonstrate that our proposed algorithm can improve the performance including network latency, energy consumption and throughput compared with traditional routing algorithms in various scales of networks. In the application of a large scale network such as $16 * 16 * 3$ mesh topology, our proposed algorithm can reduce the latency by 16.7% and the energy by 34.2% on average, compared with other traditional routing algorithms.

The rest of the paper is organized as follows: Section 2 presents previous research related to the study, Section 3 gives the problem definition and briefly reviews the adaptive routing algorithm, Section 4 describes the proposed 3D routing algorithms including inter-layer and intra-layer routing, Section 5 illustrates the simulation results and Section 6 concludes the paper.

2. Related Works

There are many works on 3D NoCs designs in the communication interconnect of 3D integration including router architecture, topology and layout, routing algorithms and network synthesis. Reference [16] proposed a partially-connected 3D crossbar structure, which provides a good tradeoff between circuit complexity and performance benefits. Reference [8] presented a layer-multiplexed (LM) 3D network architecture which replaces the one-layer-per-hop routing in a conventional 3D mesh with simpler vertical demultiplexing and multiplexing structures. Application-Specific 3D NoC topology and layout optimizations were developed in Refs. [17], [18], [19] aiming to obtain better power consumption and latency than classical topologies. Network synthesis problem was researched in Refs. [20] and [21].

As well as NoC architecture and structures, a routing algorithm is very important because it largely affects the performance in NoC applications. Designing an effective routing algorithm is a critical problem. Many kinds of routing algorithms including deterministic routing such as Dimension-Order Routing [1], and adaptive routing such as odd-even [9], turn-model [10] in 2D and 4NP-first [2] in 3D were proposed. Most of them were designed on the basis of shortest route, balance of network state, and free of deadlock and livelock [12]. An adaptive routing algorithm uses information about the network state to select better paths to deliver a packet [1]. By using the congestion information, the network can be load balanced, and the latency can be improved largely. Partially adaptive routing algorithm imposes restrictions on turns to avoid deadlocks, and consequently it can only use a subset of the minimal paths for routing [6], which gives a limitation to the path selection for effectively balancing the traffic load. Fully adaptive routing allows free usage of all available routing paths, which is more flexible and gives a better improvement of the traffic congestion. Several fully adaptive routing algorithms were researched in recent years, however some of them were based on only minimal path search [6], and others were using routing tables [4]. Besides, fewer works were addressed on

3D NoC routing. References [2] and [22] presented turn-model based partially adaptive routing algorithms in 3D NoCs, but they couldn't achieve full adaptivity due to the restrictions of routing direction in turn-model. In Ref. [3], a vertical node selection scheme in inter-layer routing was proposed, but it was static method and always considered the minimum distance from the source node. In Ref. [23], an Ant Colony Optimization based adaptive routing algorithm was applied in 3D mesh NoC, but it was also a static method, and can be used only in the system with a fixed set of tasks. Since the procedure for route searching is quite time consuming, it could't be extended to dynamic network.

In order to solve the above drawbacks, we propose a fully adaptive routing algorithm in 3D NoC with dynamic traffic patterns in which an adaptive vertical node selection method is applied using both source and destination nodes to make more flexible routing decisions.

Due to the process technology, currently available processes for TSV fabrication show relatively low yield [13]. Recent works have paid much attention on increasing the yield by improving the reliability of TSV interconnects at the expense of different levels of hardware overhead. The most popular method is hardware redundancy [26]. In Ref. [13], a redundancy based defect-tolerant technique was presented to improve the TSV yield while keeping the area increase within a modest level. Reference [27] proposed a redundant TSVs grouping technique to select good signal paths away from defective TSVs, and then improves the yield. Grouping ratio was used to control the hardware overhead increase. In Ref. [11], a scalable redundant TSV scheme was proposed to fit the failure rate of TSV for different TSV processes and bonding technologies. In this work, we don't consider altering the original hardware architecture of the 3D NoC, and focus on exploring an optimal routing algorithm under the given TSV constraint. The number of TSV is fixed as an input parameter to the system.

3. Problem Formulation

3.1 3D NoC Model

In the application of NoC design, topology, routing and flow control are three major items. The interconnection network is implemented with a collection of shared router nodes connected by shared channels, and the connection pattern of these nodes defines the network's topology. Routing determines which of the possible paths a message actually takes to transfer through the network from its source terminal to its destination terminal. Flow control dictates which messages get access to particular network resources over time [1].

In this paper, a topology of 3D mesh with limited quantity of TSVs is applied. Different from the traditional 3D Symmetric NoC architecture in which every router is a 3D router, we implement our routing algorithm on a more practical network with fixed number of 3D routers (shown in **Fig. 2** (a) [16] and (b)). The number and distribution of 3D routers are determined by the 3D TSV constraint. There are two kinds of routers in the architecture: 2D router (we also call it 2D node) with 5 bidirectional channels, which can only transfer message within the same layer, and 3D router (vertical node) with 7 bidirectional channels connecting with TSVs, which can deliver the message between different

layers. The aim of this work is to design a routing algorithm which is suitable for specific applications under the 3D TSV constraint. The algorithm can not only decrease the message transmission distance, but also balance the load of the shared network resources. It is also free of deadlock and livelock. The flit buffer based virtual-channel flow control is adopted as the network flow control scheme instead of wormhole flow control.

3.2 Adaptive Routing

An adaptive routing algorithm uses information about the network state to select a proper path among alternative paths to deliver a packet [1]. It consists of partially adaptive routing and fully adaptive routing. Partially adaptive routing enables a limited degree of adaptivity, primarily to avoid deadlock [2]. Fully adaptive routing makes a free choice of routing directions. Partially adaptive routing usually restricts packets to travel along a shortest path to the destination, while fully adaptive routing is more flexible for path selection. Traditional fully adaptive routing algorithms are usually based on routing table. For the table based routing, every router should reserve a table storing the routing relations between the input channels and the output channels, and it should be periodically updated. The lack of scalability is one of the drawbacks of the table based routings, that is, table sizes increase rapidly with NoC size, so that router (or NI) area, energy and latency increase, and possibility of deadlocks occurring becomes large [2]. In this work, a fully adaptive routing algorithm without a routing table is designed, in which the candidate path is evaluated using a routing function described in Section 4.2 by comprehensively considering the state of the neighbor nodes. The calculation process is simple and fast, which can significantly decrease the network latency.

3.3 Problem Definition

Given a 3D mesh topology and the specific application, the problem deals with selecting the proper paths for all packets between the respective source and destination, traversing the network. The objective is to improve the network performance, such as latency, energy consumption and throughput. In the system, we input the topology parameters, application parameters, communication parameters and the router architectures. The topology parameters include the mesh structure, the number and distributions of 2D and 3D routers; the application parameters describe the communications among cores, including packets size, total packet counts, source and destination positions, etc.; the communication parameters describe the application bandwidth, latency,

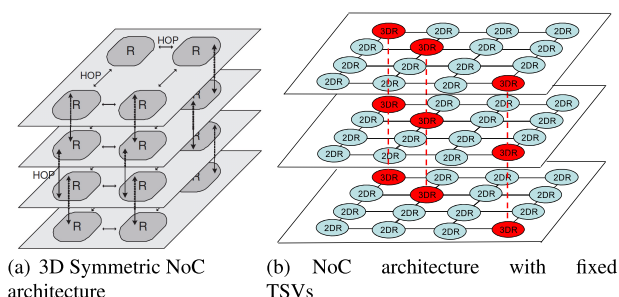


Fig. 2 3D NoC architecture.

message types etc.; and the router architectures describe the hardware structure of the applied routers. When a packet arrives at a node, the router implements the proposed routing algorithm to calculate the proper path for the next hop.

4. Proposed Algorithm

The routing algorithm is composed of two parts: vertical node assignment in inter-layer routing and 2D fully adaptive routing algorithm in intra-layer routing. The flow chart of the algorithm is illustrated in Fig. 3.

In the initialization, the router read the source/destination and traffic information of the network. Traffic information means the current queue occupancies in the downstream routers, which represents the congestion state in the network. In the algorithm, packets are firstly routed towards the destination layer. The node containing a TSV for vertical transmission is called vertical node which is 3D router. Since not every node is designed as a vertical node in the topology with limited quantity of TSVs, we must assign a proper vertical node for the packets which have to traverse multi layers. Algorithm 1 (in Section 4.1) is applied for choosing a best vertical node for ensuring the minimum transmission distance. Routing within the same layer is determined by Algorithm 2 in Section 4.2), which is a fully adaptive 2D routing algorithm considering the optimization of routing distance, network traffic condition and diversity [2] of the path selection. It

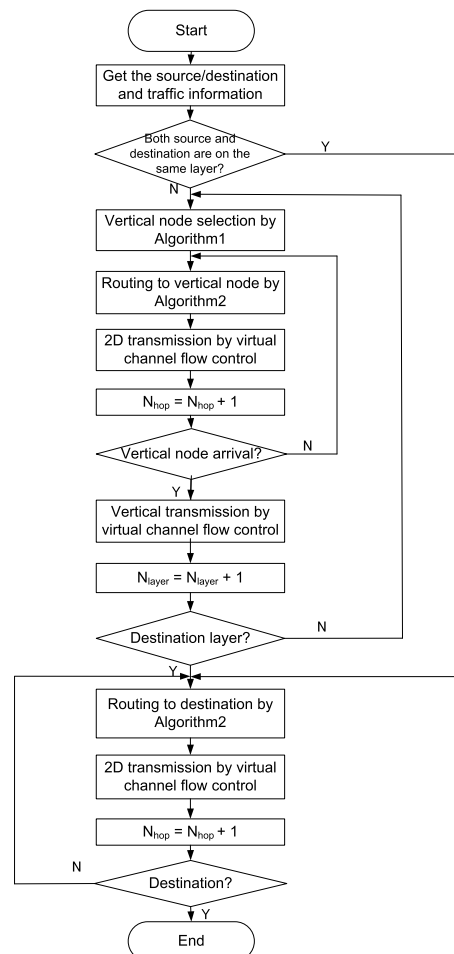


Fig. 3 Flow chart of the proposed algorithm.

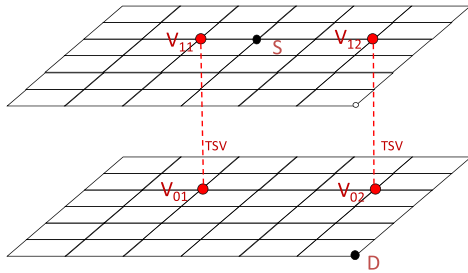


Fig. 4 Example of 3D NoC optimization.

also includes deadlock and livelock control schemes.

4.1 3D Inter-layer Routing Algorithm

The 3D inter-layer routing intends to route the packet from the source layer to the destination layer. The core of the algorithm is vertical node assignment, which is used to select vertical nodes for inter-layer transmission. Most traditional methods are based on a static algorithm in which the node nearest to the source node is selected as a candidate vertical node, considering the distance and simplicity of processing [3]. However, in some cases, it may not be the best of choice for shortening the transmission distance. Let's consider the following situation for example (shown in Fig. 4). In this two-layer network, S and D are the source node and the destination node respectively, and V_{01} , V_{02} , V_{11} and V_{12} are four vertical nodes. According to the static method, V_{11} will be selected as the candidate vertical node due to its minimum distance from S . But in fact, V_{12} is better than V_{11} because the total transmission distance between S and D can be decreased.

To overcome the disadvantage of the traditional method, we propose a dynamic vertical node selection, in which the vertical node for inter-layer transmission in each layer is adaptively selected according to the source and destination node in every routing decision. We use S and D to represent the source node and destination node respectively, and their 3D coordinates are defined by $S(x_S, y_S, z_S)$ and $D(x_D, y_D, z_D)$. V means the set of vertical nodes v_i , where $i = 1, 2, \dots, N$. D' denotes the corresponding destination in the same layer with S , that is, $D'(x_D, y_D, z_S)$, and D'' means the corresponding destination in the adjacent layer to S 's layer, that is, $D''(x_D, y_D, z_{S \pm 1})$. The selected vertical node in V is denoted as v' . The pseudo code of vertical node assignment algorithm is shown in Algorithm 1 below. There are two cases in this algorithm:

- (case-1) D and S are in adjacent layers;
- (case-2) D and S are not in adjacent layers.

In case-1, if there are more than one v_i in the shortest path area from S to D' , which is defined by the minimum rectangle including S and D' , the algorithm will randomly select one v_i in the shortest path area; otherwise, it will select the v_i which has the minimum distance between the S and D' (the distance of (S to v_i) + (v_i to D') is minimum). In case-2, if there are more than one v_i in the shortest path area from S to D' , the algorithm will select the v_i nearest to S in the shortest path area (this ensures a high diversity for the selection in the next layer); otherwise, select the v_i which has the minimum distance between S and D' . When v' is determined, the packet is routed to the next layer to S through v' . If case-2 is applied, which means the packet doesn't

Algorithm 1 Vertical node assignment in inter-layer routing

Input: $S(x_S, y_S, z_S)$, $D(x_D, y_D, z_D)$, $V = \{v_i\}$

Output: Selected vertical node v' in V

```

begin
  if  $|z_S - z_D| = 0$  then
    return;
  if  $|z_S - z_D| = 1$  then
    begin
      set  $D' = (x_D, y_D, z_S)$ ;
      if there is more than one  $v_i$  as a candidate
        between  $S$  and  $D'$  then
        randomly select  $v'$  from  $V$  in the shortest path area;
      else
        select  $v' = v_i$  nearest to  $S$  and  $D'$ ;
    end;
  else if  $|z_S - z_D| \geq 2$  then
    begin
      set  $D' = (x_D, y_D, z_S)$ ;
      if there is more than one  $v_i$  between  $S$  and  $D'$  then
        select  $v' = v_i$  nearest to  $S$ ;
      else
        select  $v' = v_i$  nearest to  $S$  and  $D'$ ;
    end;
end
    
```

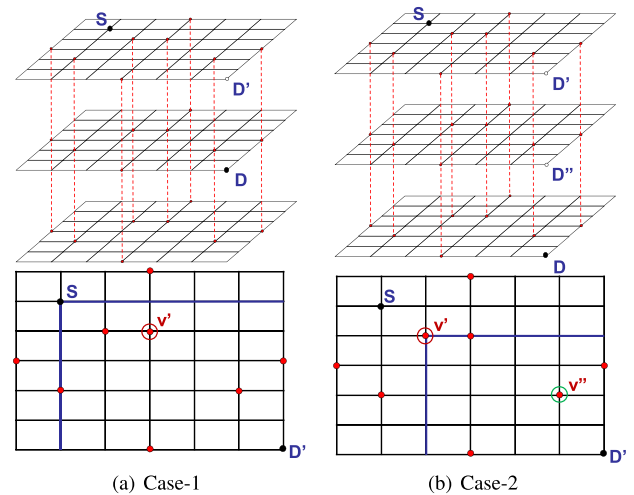


Fig. 5 Example of Algorithm 1 in $7 * 7 * 3$ mesh NoC.

reach the destination layer through one round application, then in the next layer, v' is used as the new S , and D'' is considered as the new D' , and the algorithm is applied again. According to the locations of D and S , we use two different methods to select the vertical nodes. Because, in case-2, if we first select the node nearest to S , the optional area (shortest path area) for the next layer will be maximized, which gives a high diversity for the selection in the next layer. In case-1, there is no such need, so we use a random selection scheme to balance the traffic load.

An example of Algorithm 1 in $7 * 7 * 3$ mesh NoC is shown in Fig. 4. Figure 5 (a) is an example of case-1. Since there are six v_i s in the shortest path area of S and D' , it randomly selects one as the objective vertical node. Figure 5 (b) is an example of case-2. In the first layer, it selects the v_i nearest to S , and then in the second layer, it randomly selects one in the shortest path area between v' and D'' .

4.2 2D Intra-layer Routing Algorithm

The intra-layer routing is a 2D fully adaptive routing algorithm, in which the route of the next hop is determined by distance, traffic state and path diversity together. It also includes livelock

Algorithm 2 Fully adaptive routing in 2D layer

Input: $C(x, y)$, $D(x, y)$, Free buffer depth of C_i , T_0 , N_0 , N_1
Output: routing path

begin

 Calculate N_s ;

 set $N_m = 0$;

for each neighbor C_i of C **do**

 calculate $f_i = \alpha \cdot l_i + \beta \cdot s_i + \gamma \cdot d_i$;

 set $C = C_i$ s.t. $\max f_i$
if C is not in shortest path **then**
 $N_m = N_m + 1$;

end
if $N_m = N_0$ **then**

 set $\alpha = 1, \beta = \gamma = 0$;

end

Data transmission using virtual channel flow control;

while (header is not in C) **do**
 $T_1 = T_1 + 1$;

if $T_1 > T_0$ **then**

DISHA deadlock recovery;

end

 add C to Path;

end

avoidance and deadlock recovery schemes. We denote C , D and C_i as the current node, destination node and the neighbor node of C in direction i ($i = 1$: North, 2: South, 3: East, 4: West in 2D) respectively. The pseudo code of 2D intra-layer routing algorithm is illustrated in Algorithm 2.

In this algorithm, we use the following objective function to select the best node among the adjacent four nodes.

$$f_i = \alpha \cdot l_i + \beta \cdot s_i + \gamma \cdot d_i, \quad i = 1, 2, 3, 4 \quad (1)$$

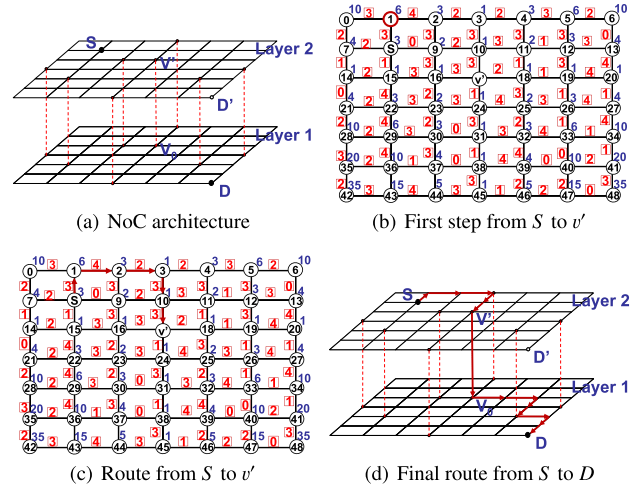
This function is used to evaluate the routing condition of each neighbor node, where l_i denotes the distance (number of hops) from C_i to D , s_i denotes each current traffic state, d_i represents the path diversity in direction i , and α, β and γ represent the weights for the three objectives. Usually we set $\alpha > \beta > \gamma$, which means that distance has the highest weight in the three objectives. Each objective is normalized to the average value of the current node as follows.

- $l_i = l_0 / l_{i1}$, where l_{i1} and l_0 denote the minimum number of hops from C_i to D , and C to D respectively.
- $s_i = s_{i1} / s_0$, where s_{i1} and s_0 denote the free buffer size and the total buffer size of C_i respectively.
- $d_i = d_{i1} / d_0$, where d_{i1} and d_0 denote the diversity number of C_i and C respectively. The diversity number is defined as the number of available shortest paths from the current node to the destination node [2].

The node C_i with the maximum function value f_i will be selected as the next transmitting node on the routing path.

To avoid livelock, we use a small amount of state in packet to record N_m which is the number of times the packet has been misrouted. Once N_m reaches a threshold N_0 , no more misrouting is allowed [1]. N_0 can be calculated as $N_0 = N_s + N_1$, where N_s denotes the number of hops in the shortest path from S to D , and N_1 is given as a constant value, meaning the number of allowed misrouting hops.

The utilization of traffic state to make routing decisions decreases the probability of deadlock occurrence. When deadlock happens, we use a deadlock recovery scheme DISHA [5]. In this case, each router is equipped with an extra flit buffer to store the


Fig. 6 Example of the routing algorithm in 2 layers.

header flit of one of the deadlock packets [7]. T_1 is attached to the 2D router to keep track of the number of clock cycles. If it is unable to send out the header, T_1 is incremented. When T_1 reaches a threshold T_0 , it is determined as deadlock.

An example of the proposed routing algorithm is shown in Fig. 6. In this example, a packet is injected in S and is going to D which is in a different layer. The algorithm firstly assigns a vertical node v' for a transmission to the layer 1 by using Algorithm 1, then the packet is routed from S ($= C$) to v' ($= D$) using Algorithm 2. After getting to v' , a transmission from v' to v_0 is implemented and finally it is routed from v_0 ($= C$) to D by use of Algorithm 2 again. Figures 6 (b) and (c) show an example of routing from S to v' . The number attached beside each node in the figure is the diversity number, and the value on each edge is the free buffer size. We set $\alpha = 0.5, \beta = 0.4, \gamma = 0.1$, and the buffer size is 4 flits. Firstly, a current node $C = S, l_0 = 3, s_0 = 4$, and $d_0 = 3$. We calculate the function value of each neighbor node, that is f_1 at node1, f_2 node7, f_3 node15, and f_4 node9, and get $f_1 = 0.875, f_2 = 0.708, f_3 = 0.858, f_4 = 0.817$. By comparing the function value, the node 1 is selected as the next hop node on path (Fig. 6 (b)). In the same way, the route from S to v' can be determined as $S \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 10 \rightarrow v'$, as shown in Fig. 6 (c). Figure 6 (d) shows the final route from S to D by using our proposed routing algorithm.

5. Experimental Results and Analysis

We evaluate the performance of the proposed routing algorithm by using an open source simulator Noxim [24], which is developed using SystemC and supports cycle-accurate simulation. Noxim is designed for analysis and evaluation of a large set of quality indices including latency, energy consumption, throughput, etc. with various configurable parameters. The simulator is modified to implement a 3D mesh NoC and extended to support our proposed routing algorithm. We use its internal performance model to estimate the network latency, energy consumption and throughput.

To test the effectiveness of the proposed 2D fully adaptive routing algorithm, we first implement our routing algorithm in the 2D dimension, and compare it with the traditional deterministic XY

routing algorithm, and adaptive odd-even [9] and west-first routing algorithms. To achieve high yield, the number of TSVs used in a 3D IC is usually restricted below some threshold value. For any process, there is a clear upper bound on the number of TSVs over which the yield decreases rapidly [20]. In our experiments, we first test our proposed routing algorithm with a given quantity and distribution of TSVs under the TSV constraint, and then implement it with a variation of quantity of TSVs, which shows the impact of a tighter TSV constraint on the performance of the network. To test the effectiveness of the proposed 3D vertical node selection algorithm, it is compared with a static vertical node selection algorithm [3]. To test the effectiveness of the proposed 3D routing algorithm, we combine the classical 2D routing algorithms including with the static vertical node selection algorithm (we call them classical 3D routing algorithms), and compare our proposed algorithm with these classical 3D routing algorithms.

We design six different experiments considering different parameters and assessing their impact on the system performance. Four performance indices including average latency, maximum latency, energy consumption and throughput are evaluated and input to Matlab [25] for statistical analysis. The first experiment in Section 5.1 is designed to test our algorithm in the 2D application. The following three experiments in Sections 5.2 to 5.4 use different network size to test our routing algorithm, and the results are compared with a deterministic XY routing algorithm with static vertical node selection and an adaptive odd-even routing algorithm with static vertical node selection. The fifth experiment in Section 5.5 is to evaluate our algorithm in different traffic patterns, and it is compared with XY, odd-even and west-first routing algorithms with static vertical node selection. The last experiment in Section 5.6 is to illustrate the impact of TSV quantity and distribution.

The router employs a virtual channel flow control, each physical channel consists of two virtual channels, and each flit buffer has a capacity of four flits. The simulator generates a series of packets with the size from 2 flits to 10 flits, and the packet injection rate is varying from 0.1 to 0.95. To determine the values of the weight parameters α , β and γ , we implement several simulations with different combinations beforehand for each topology with a given traffic pattern, and then choose the proper parameter set which can achieve the best system performance. We can use this parameter set for the network traffic conditions, because the network conditions will not make a great impact on the weight parameters. In other words, even if the determined parameter set

may not be the best choice, the result is reasonably good. The five combinations of the parameters are shown in **Table 1**. The number and the distribution of vertical nodes are given as an input to the system. According to Ref. [20], a stricter constraint on the TSV number will lead to higher power consumption and latency. For our experiments, it's hard to get the correlation between the real chip yield and TSV number. We just use the simulation tool to simulate the system performance with different number and distributions of TSVs, and select one that generates reasonable latency and power consumption. In Sections 5.2 to 5.5, we set the TSV threshold as 60% of the total number of nodes, with a random distribution among the mesh network. The vertical nodes appear in pairs, which means if there is a vertical node in one layer, we can find the corresponding one in the adjacent layer. The number of vertical nodes and the distributions in the bottom layer in experiments from Sections 5.2 to 5.4 are shown in **Table 2** (The distribution in Section 5.5 is the same as Section 5.3). "Node ID" is the number of ID in the 3D NoC architecture. In real applications, the TSV threshold can be adjusted according to yield and performance requirement, and our proposed routing algorithm can also be applied. In Section 5.6, we illustrate the impact on different TSV constraint strategies.

5.1 Results on 16 * 16 * 1 Mesh NoC

To evaluate the 2D intra-layer routing algorithm, we first implement the simulations on a 2D 16 * 16 mesh network with random traffic pattern. The comparisons including (a) average latency, (b) maximum latency, (c) energy consumption, and (d) throughput are illustrated in **Fig. 7**.

By comparing with the classical 2D routing algorithms, our proposed one can improve the latency, energy consumption, and throughput significantly, which shows the 2D intra-layer routing part of the proposed algorithm is effective to improve the network performance. As a result, our proposed routing algorithm can also be considered as a promising solution in the 2D NoC.

5.2 Results on 4 * 4 * 3 mesh NoC

In the 3D application, we first implement the simulations on a

Table 1 Weight parameters in the experiments.

	Case 1	Case 2	Case 3	Case 4	Case 5
α	0.7	0.6	0.8	0.65	0.7
β	0.2	0.3	0.15	0.2	0.2
γ	0.1	0.1	0.05	0.15	0.1

Table 2 Number of vertical nodes and the distributions in the bottom layer.

	Number of vertical nodes	Vertical node ID
4 * 4 * 3 mesh in Section 5.2	30	1, 3, 6, 8, 10, 11, 13, 14, 15, 16
8 * 8 * 4 mesh in Section 5.3	152	2, 4, 5, 7, 8, 9, 11, 12, 16, 18, 19, 21, 23, 24, 25, 27, 29, 32, 34, 36, 37, 39, 40, 41, 43, 45, 46, 47, 50, 51, 52, 54, 56, 58, 59, 60, 62, 64
16 * 16 * 3 mesh in Section 5.4	462	1, 4, 5, 6, 9, 10, 11, 13, 15, 16, 18, 20, 21, 22, 25, 29, 30, 31, 32, 33, 34, 37, 38, 40, 42, 43, 45, 48, 49, 50, 54, 55, 58, 60, 62, 63, 64, 66, 67, 69, 70, 72, 75, 76, 77, 80, 81, 84, 85, 87, 89, 91, 92, 93, 95, 97, 100, 101, 103, 104, 106, 108, 110, 113, 114, 116, 119, 120, 123, 126, 128, 130, 131, 133, 134, 136, 137, 138, 140, 142, 143, 145, 146, 147, 148, 150, 151, 153, 155, 156, 158, 159, 160, 161, 162, 166, 168, 170, 173, 174, 177, 178, 179, 180, 182, 183, 185, 188, 189, 191, 192, 194, 196, 199, 200, 203, 204, 206, 207, 210, 211, 212, 214, 215, 218, 220, 221, 222, 223, 225, 226, 227, 229, 230, 232, 233, 235, 237, 238, 239, 240, 242, 243, 244, 245, 247, 248, 249, 250, 252, 253, 254, 255, 256

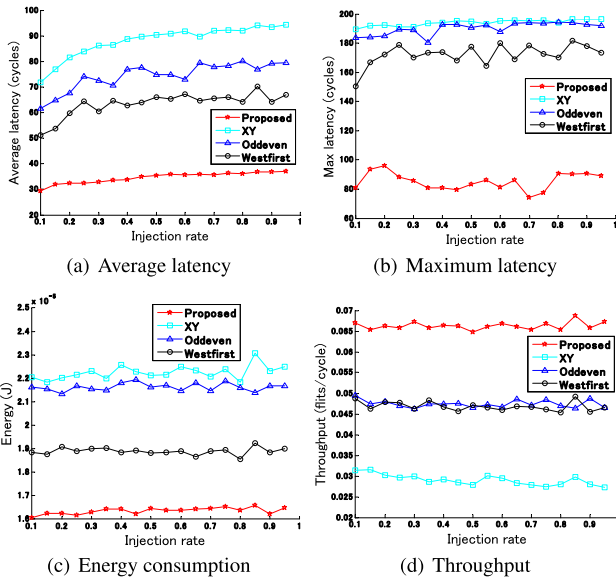


Fig. 7 Experimental results on 16 * 16 * 1 mesh NoC.

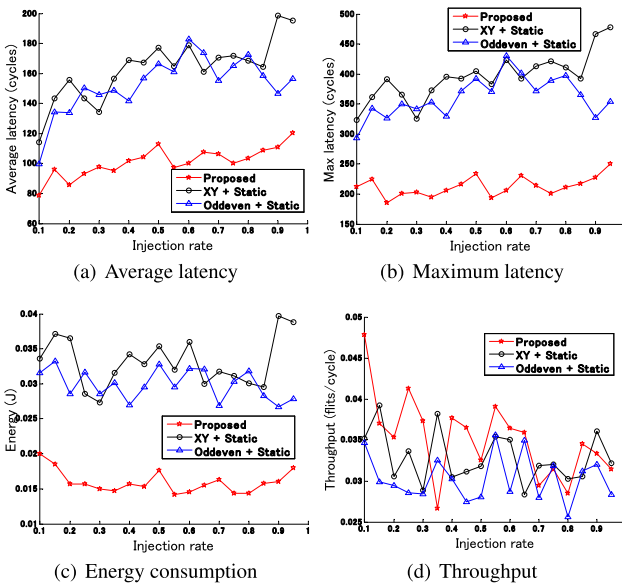


Fig. 8 Experimental results on 4 * 4 * 3 mesh NoC.

small scale 3D NoC, that is 4 * 4 * 3 mesh network with random traffic pattern. We take 1000 runs and record the average values for evaluation. The comparisons are illustrated in Fig. 8.

From these figures we can find our proposed routing algorithm can improve the latency and energy consumption a great deal and improve the throughput to a certain extent. It also proves that our proposed algorithm is effective to improve the network performance in small scale 3D NoC applications.

5.3 Results on 8 * 8 * 4 mesh NoC

We then implement the simulations on a medium scale 3D NoC, that is 8 * 8 * 4 mesh network with random traffic pattern. The comparisons are illustrated in Fig. 9.

The results show that our proposed algorithm outperforms other traditional routing algorithms on considering the system performance. We can also find that the performance improvement is more remarkable when the number of layers increases. Our proposed routing algorithm can be also applied in the network with more than four layers.

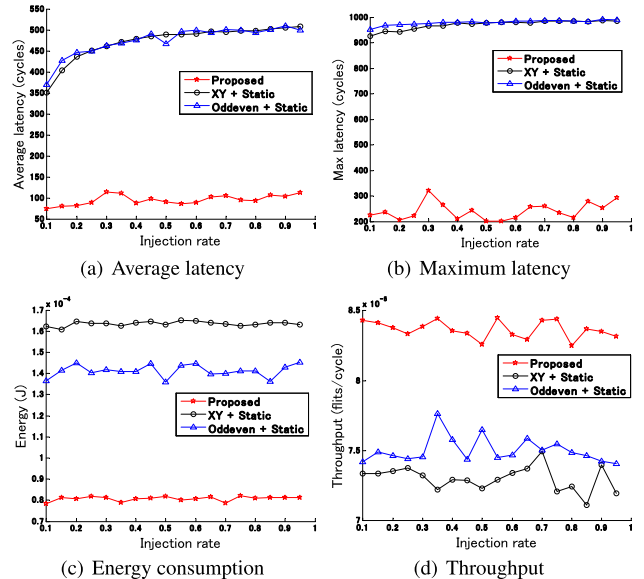


Fig. 9 Experimental results on 8 * 8 * 4 mesh NoC.

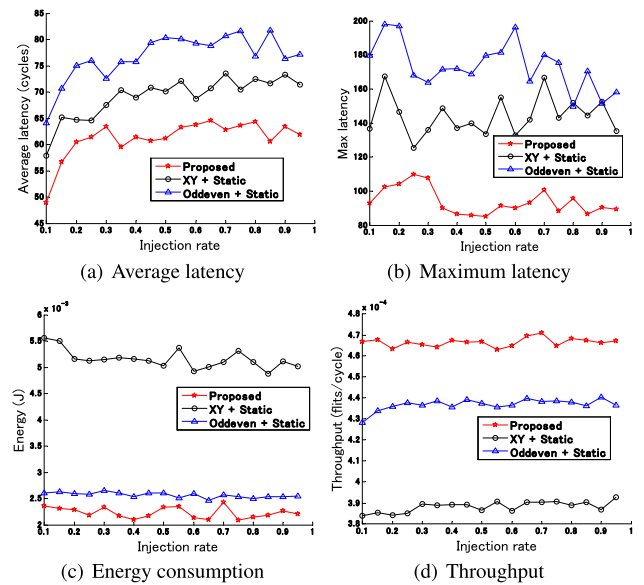


Fig. 10 Experimental results on 16 * 16 * 3 mesh NoC.

5.4 Results on 16 * 16 * 3 mesh NoC

In the third experiment, the simulations are implemented on a larger scale network: 16 * 16 * 3 mesh NoC with random traffic pattern. The comparisons are illustrated in Fig. 10.

In this experiment, our proposed algorithm achieves good result by comparing with other algorithms. On average, our proposed algorithm reduce the latency 16.7% and energy 34.2% compared with other algorithms. The results also illustrate our proposed algorithm is effective in the application of larger scale NoCs.

5.5 Results at Other Traffic Patterns

In this experiment, to evaluate the effectiveness of the proposed algorithm for the different traffic patterns, another two well-known traffic patterns shuffle and transpose are used on an 8*8*4 mesh NoC. The comparisons at shuffle pattern and at transpose pattern are illustrated in Fig. 11 and Fig. 12 respectively.

In these experiments, our proposed routing algorithm shows higher throughput and lower energy compared with other algo-

gorithms. As a result, our proposed algorithm is effective in both energy and latency when the network traffic varies.

5.6 Results at Different TSV Quantity

Since the number of TSVs affects the TSV connection yield, the constraint of TSV numbers must be considered when design-

ing a 3D NoC topology. In this experiment, we will test the impact of TSV number on our routing algorithm. The simulation is implemented on a $4 \times 4 \times 2$ mesh NoC with the TSV number varying from 1 to 16. The implementation results are shown in Fig. 13.

From the results, we can find that energy consumption and network throughput are remarkably affected by the increasing of TSV quantity, while latency is not affected by it.

6. Conclusions

In this paper, we proposed an energy and latency effective fully adaptive routing algorithm for 3D NoC with fixed TSVs. The algorithm used a novel vertical node selection method to adaptively assign a proper vertical node for vertical transmission and a fully adaptive 2D routing algorithm in which transmission distance, traffic state, and route diversity are considered comprehensively. To summarize the comparison results in different scales of networks, Table 3 illustrates the average result at random traffic within the range of the injection rate from 0.1 to 1.0. By comparing with other classical routing algorithms on different network scales, our proposed algorithm achieved better performance considering the network throughput, latency, and energy consumption simultaneously. Especially for medium sized NoCs, such as $8 \times 8 \times 4$ network, the proposed algorithm can achieve over 50% latency improvement, and the energy and throughput improvement is also remarkable. By comparing with XY routing and

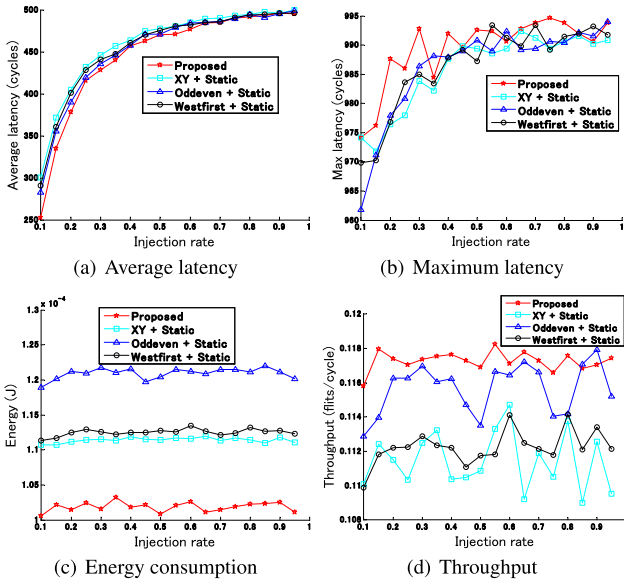


Fig. 11 Experimental results at shuffle pattern.

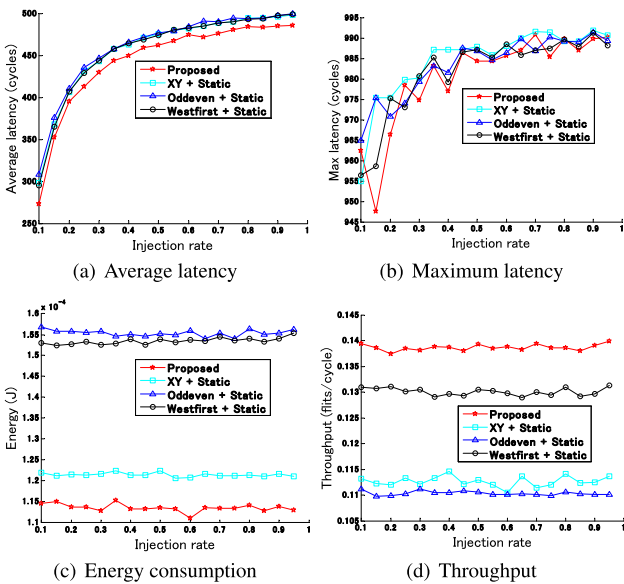


Fig. 12 Experimental results at transpose pattern.

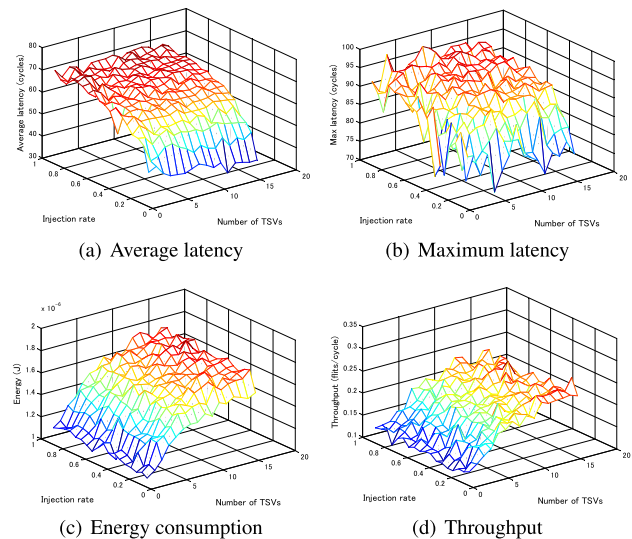


Fig. 13 Simulation results with different TSVs.

Table 3 Summary of comparison.

Item	Ave. latency (cycles)			Energy (J)			Throughput (flits/cycle)		
	(1)	(2)	Proposed	(1)	(2)	Proposed	(1)	(2)	Proposed
$4 \times 4 \times 3$	159.72	149.89	101.11 (1) 36.70% (2) 32.54%	0.033	0.029	0.016 (1) 51.02% (2) 45.47%	0.033	0.03	0.035 (1) 5.32% (2) 15.25%
$8 \times 8 \times 4$	467.22	465	101.94 (1) 78.18% (2) 78.08%	1.61×10^{-4}	1.41×10^{-4}	0.82×10^{-4} (1) 49.03% (2) 41.82%	7.35×10^{-6}	7.51×10^{-6}	8.37×10^{-6} (1) 13.98% (2) 11.48%
$16 \times 16 \times 3$	68.11	75.89	59.83 (1) 12.15% (2) 21.16%	5.17×10^{-3}	2.57×10^{-3}	2.26×10^{-3} (1) 56.34% (2) 12.01%	3.89×10^{-4}	4.38×10^{-4}	4.67×10^{-4} (1) 19.88% (2) 6.54%

(1) XY + Static, (2) Oddeven + Static, % means the ratio of improvement to (1) or (2)

static vertical node selection, the proposed algorithm achieves an average of 42.34% latency improvement, 52.13% energy improvement, and 11.09% throughput improvement on implementation on different scales of network. By comparing with odd-even routing and static vertical node selection, the proposed algorithm achieves an average of 43.93% latency improvement, 33.1% energy improvement, and 11.04% throughput improvement on implementation on different scales of network. On implementation on 2D network and different traffic patterns, our proposed algorithm is also effective to improve the system performance.

Acknowledgments This research was partly supported by JSPS KAKENHI 23500069.

References

[1] Dally, W. and Towles, B.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann Publishers Inc., San Francisco, CA (2003).

[2] Pasricha, S. and Zou, Y.: A low overhead fault tolerant routing scheme for 3-D networks-on-chip, *Int. Symp. Quality Electron. Design*, pp.1–8 (2011).

[3] Rusu, C., Anghel, L. and Avresky, D.: Adaptive inter-layer message routing in 3D networks-on-chip, *Microprocessors and Microsystems*, Vol.35, No.7, pp.613–631 (Oct. 2011).

[4] Feng, C., Zhang, M., Li, J., Jiang, J. and Jantsch, A.: A Low-Overhead Fault-Aware Deflection Routing Algorithm for 3D Network-on-Chip, *Proc. 2011 IEEE Computer Society Annual Symposium*, pp.19–24 (2011).

[5] Anjan, K.V. and Pinkston, T.: An efficient fully adaptive deadlock recovery scheme: DISHA, *Proc. 22nd annual international symposium on Computer architecture*, pp.201–210 (June 1995).

[6] Wen-Chung, T. et al.: TM-FAR: Turn-Model Based Fully Adaptive Routing for Networks on Chip, *Proc. VLSI-SoC*, pp.19–24 (2010).

[7] Patooghy, A. and Miremadi, S.G.: Complement routing: A methodology to design reliable routing algorithm for Network on Chips, *Microprocessors and Microsystems 34*, pp.163–173 (2010).

[8] Ramanujam, R.S. and Lin, B.: A Layer-Multiplexed 3D On-Chip Network Architecture, *IEEE Embedded Systems Letters*, Vol.1, No.2, pp.50–55 (2009).

[9] Chiu, G.: The Odd-Even Turn Model for Adaptive Routing, *IEEE Trans. Parallel and Distributed System*, Vol.11, pp.729–738 (July 2000).

[10] Glass, C.J. and Ni, L.M.: The Turn model for Adaptive Routing, *Proc. Symp. Computer Architecture*, pp.278–287 (May 1992).

[11] Hsieh, A.C. and KhursheedHwang, T.T.: TSV redundancy: Architecture and design issues in 3D IC, *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, Vol.20, No.4, pp.711–722 (2012).

[12] Ramanujam, R.S. and Lin, B.: Near-optimal oblivious routing on three dimensional mesh networks, *Proc. IEEE Int. Conf. Comp. Design, Lake Tahoe, CA* (2008).

[13] Loi, I., Mitra, S., Lee, T.H., Fujita, S. and Benini, L.: A low-overhead fault tolerance scheme for TSV-based 3D network on chip links, *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp.598–602 (2008).

[14] Loi, I., Angiolini, F. and Benini, L.: Supporting vertical links for 3D networks on chip: toward an automated design and analysis flow, *Proc. Nanonets* (2007).

[15] Miyakawa, N.: A 3D prototyping chip based on a wafer-level stacking technology, *Proc. 2009 Asia and South Pacific Design Automation Conference (ASPDAC)*, pp.416–420 (2009).

[16] Kim, J. et al.: A novel dimensionally-decomposed router for on-chip communication in 3D architectures, *Proc. Int. Symp. Comput. Architecture*, pp.138–149 (2007).

[17] Zhou, P.Q., Yuh, P.H. and Sapatnekar, S.S.: Application-specific 3D Network-on-Chip design using simulated allocation, *Proc. Design Automation Conference (DAC)*, pp.517–522 (2010).

[18] Yan, S. and Lin, B.: Design of application-specific 3D networks-on-chip architectures, *Proc. 26th Int. Conf. Comput. Des.*, pp.142–149 (2008).

[19] Jiang, X., Zhang, R. and Watanabe, T.: An Efficient Algorithm for 3D NoC Architecture Optimization, *IP SJ Trans. System LSI Design Methodology*, Vol.6, pp.34–41 (2011).

[20] Seiculescu, C., Murali, S., Benini, L. and De Micheli, G.: SunFloor 3D: A tool for networks on chip topology synthesis for 3D systems on chip, *Proc. DATE*, pp.9–14 (2009).

[21] Murali, S., Seiculescu, C., Benini, L. and De Micheli, G.: Synthesis of networks on chips for 3D systems on chips, *Proc. the 2009 Asia and South Pacific Design Automation Conference*, pp.19–22 (2009).

[22] Ying, H., Jaiswal, A. and Hofmann, K.: Deadlock-Free Routing Algorithms for 3-Dimension Networks-on-Chip with Reduced Vertical Channel Density Topologies, *HPCS-DRNoC* (2012).

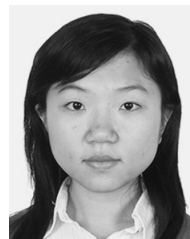
[23] Junior, L.S., Nedjah, N. and de Macedo Mourelle, L.: ACO approach in static routing for network-on-chips with 3D mesh topology, *In Circuits and Systems (LASCAS) 2013 IEEE 4th Latin American Symposium on IEEE*, pp.1–4 (2013).

[24] Noxim, available from (<http://noxim.sourceforge.net/>).

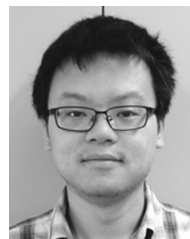
[25] MATLAB, available from (<http://www.mathworks.com/products/matlab/>).

[26] Pasricha, S.: Exploring serial vertical interconnects for 3D ICs, *Proc. 46th Annual Design Automation Conference (DAC '09)*, pp.581–586 (2009).

[27] Zhao, Y., Khursheed, S. and Al-Hashimi, B.M.: Cost-Effective TSV Grouping for Yield Improvement of 3D-ICs, *Proc. 2011 Asian Test Symposium*, pp.201–206 (2011).



Xin Jiang was born in LiaoNing, China, 1981. In 2006, she received her M.S. degree in Electronic Engineering in Shanghai Marine University. She is currently working toward a Dr.Eng. degree in Graduate School of IPS, from Waseda University. Her current research interests include optimization algorithms in Electronics DA designs. She is a member of IEICE.



Lian Zeng was born in MianYang, China, on July, 1990. He received his B.E. degree in Software Engineering in 2011, from Sichuan University. In 2013, he received his M.S. degree and be currently working toward a Ph.D. degree in Graduate School of Information, Productions and Systems, from Waseda University. He is a member of IEICE.



Takahiro Watanabe was born in Ube, Japan, 1950. He received his B.E. and M.E. in Electrical Engineering from Yamaguchi University, and Dr.Eng. from Tohoku University. In 1979, he joined R&D Center of TOSHIBA Corp. In August 1990, he joined Yamaguchi University, the Department of CSSE, and in April

2003, he moved to Waseda University, Graduate School of IPS. His current research interests are EDA algorithm, Processor, and NoC, FPGA and their applications. He is a member of IEICE, IPSJ and IEEE.

(Recommended by Associate Editor: Yasuhiro Takashima)