

Floorplan Optimization of Fat-Tree Based Networks-on-Chip for Chip Multiprocessors

Zhehui Wang, *Student Member, IEEE*, Jiang Xu, *Member, IEEE*,
Xiaowen Wu, *Student Member, IEEE*, Yaoyao Ye, *Student Member, IEEE*,
Wei Zhang, *Member, IEEE*, Mahdi Nikdast, *Student Member, IEEE*,
Xuan Wang, *Student Member, IEEE*, and Zhe Wang, *Student Member, IEEE*

Abstract—Chip multiprocessor (CMP) is becoming increasingly popular in the processor industry. Efficient network-on-chip (NoC) that has similar performance to the processor cores is important in CMP design. Fat-tree based on-chip network has many advantages over traditional mesh or torus based networks in terms of throughput, power efficiency and latency. It has a bright future in the development of CMP. However, the floorplan design of the fat-tree based NoC is very challenging because of the complexity of topology. There are a large number of crossings and long interconnects, which cause severe performance degradation in the network. In electronic NoCs, the parasitic capacitance and inductance will be significant. In optical ones, large crosstalk noise and power loss will be introduced. The novel contribution of this paper is to propose a method to optimize the fat-tree floorplan, which can effectively reduce the number of crossings and minimize the interconnect length. Two types of floorplans are proposed, which could be applied to fat-tree based networks of arbitrary size. Compared with the traditional one, our floorplans could reduce more than 87% of the crossings. Since the traversal distance for signals is related to the aspect ratio of the processor cores, we also present a method to calculate the optimum aspect ratio of the processor cores to minimize the traversal distance.

Index Terms—On-chip interconnection networks, physical structures, optimization

1 INTRODUCTION

CHIP multiprocessor (CMP) is one of the modern technologies that can increase the processor performance effectively [1]. In CMP, instructions are parallel processed in multiple cores, and each core does parts of work. The cores share large on-chip caches and communicate with each other through on-chip interconnects [2]. Network-on-chip (NoC) is one of the communication systems, and plays an important role in CMP [3]. NoC has many design criteria that need to be considered [4]. Traditional NoC uses electronic interconnects. Optical network-on-chip (ONoC), which uses optical interconnects shows better throughput, delay and power efficiency than traditional NoC [5] [6]. The topology and floorplan are the two main concerns in NoC design [7].

NoCs can be classified by different topologies. Regular topologies used by NoCs include mesh, torus, ring (one-dimension torus), fat-tree, etc [8]. The fat-tree topology looks like a complete binary tree. In fat-tree, more than one interconnect exist between a node and its parent. This feature increases the throughput of the fat-tree based network [9] [10], making it one of the best routing network for multi-computer system. The fat-tree based network was proposed by Leiserson [11]. Many multi-computer systems use fat-tree based network [12], such as CM-5 [13] and Roadrunner [14]. Fat-tree topology is also a good choice for NoC because

of its high throughput characteristics [15]. SPIN [16] is one of the high performance fat-tree based electronic NoCs. Fat-tree based NoC is found to be useful not only in traditional electronic NoC but also in the emerging ONoC. Xu *et al.* [17] proposed a fat-tree based ONoC, which takes the advantage of both ONoC and fat-tree topology.

The floorplan design for large-scale fat-tree based ONoC is not easy. Optical interconnects are fabricated on a single layer, and the interconnects go across each other, which result in a large number of crossings. Since each crossing attenuates optical signal by 0.12 dB [18], a lot of power will be wasted. Moreover, the crosstalk noise introduced by crossings will decrease the signal-to-noise ratio (SNR), and degrade the network performance [19]. Floorplan of electronic NoC using fat-tree topology also has similar disadvantages. Fat-tree based NoC need to implement more vias, and it consumes more power than mesh or torus based NoC [20]. Its power and area cost increases quickly when it scales [21]. Besides, the global wire delay in fat-tree based NoC is largely related to the parasitic capacitance and inductance. Supply voltage reduction will also make the transmission of signals in long wires unreliable [15]. The power use of the global interconnects in the whole chip is a large proportion of the overall power consumption [22] [23].

The fat-tree based ONoC has many crossings. If the network size increases, the number of crossings and power loss will increase quickly. A method to optimize floorplan is required, so that the power consumption and delay of the fat-tree based NoC can

Z. Wang, J. Xu, X. Wu, Y. Ye, M. Nikdast, X. Wang, Z. Wang are with the Hong Kong University of Science and Technology, Hong Kong, China. E-mail: {zhehui, jiang.xu}@ust.hk
W. Zhang is with Nanyang Technological University, Singapore. E-mail: zhangwei@ntu.edu.sg

be reduced by decreasing the number of crossings and interconnect lengths. We propose an method to optimize the floorplan of the fat-tree based NoC. The floorplan is improved while the fat-tree topology is maintained. Compared with the traditional design, our optimized floorplans can greatly reduce the number of crossings at any network size. As a result, we can implement large-scale fat-tree based NoCs with small power loss. Two types of floorplan have been proposed. Both of them have small values on router-to-router interconnect length or core-to-core traversal distance. We can choose one of the two floorplans based on their properties.

The rest of the paper is organized as follows. Section 2 gives a survey of related work on floorplan design of fat-tree based NoC. Section 3 introduces the network architecture. Section 4 proposes the methods to reduce the number of crossings. Section 5 discusses methods to design the floorplan with minimum link length. Section 6 shows quantitative analysis and comparison. Conclusions are drawn in Section 7.

2 RELATED WORK

Some work has studied the fat-tree based NoCs and also proposed the floorplan designs. The proposed floorplans have many characteristics in common. Duto *et al.* proposed a reduced unidirectional fat-tree network [24]. Ismail *et al.* proposed a high throughput butterfly fat-tree network [25]. Ivanov *et al.* proposed a scalable communication-centric network [26]. Akbar *et al.* proposed an extended-butterfly fat-tree interconnection architecture [27]. Viswanath *et al.* proposed a fat-tree based network for caches [28]. The floorplan design in these networks looks like the letter "H". It is currently the most popular one used in the floorplan design of fat-tree based NoCs.

Some work has focused on alternative designs of floorplans or layouts in NoC. Their goal is to improve the network performance by changing the floorplan design. Batten *et al.* proposed a point-to-point layout and serpentine layout for 8-ary 2-fly butterfly based NoC [29]. They implemented a group of waveguides as the channels between two stages of routers. This layout could also be used in crossbar based NoCs. Marculescu *et al.* proposed mathematical models to evaluate the performance of NoC [30] [31]. Dally *et al.* proposed a flattened butterfly NoC [32]. This topology uses high-radix routers, which looks like 4-ary fat-tree topology. After exploring the network in two-dimensional planar layout, they added bypass channels, so that the power loss and latency can be reduced. Hsu *et al.* proposed a fat H-Tree NoC [33]. They placed several fat-tree networks on the same planar space, and made shortcuts among those networks. Such floorplan design can also be used in three-dimensional IC designs, and increases the network throughput. The aforementioned work improved the network architecture by changing its topology and

routing algorithms. Their basic floorplan design still use the traditional "H-tree" pattern, which has a large number of crossings. This paper discuss the method to optimize the traditionally used "H-tree" floorplan, so that the number of crossings can be reduced.

Some work has explored methods for floorplan design in NoC. General methods were proposed, which could be used for different topologies. De Micheli *et al.* proposed a floorplan design tool for electronic NoCs [34]. Their goal is to minimize the interconnect lengths and make it compatible with routing algorithms. Chao *et al.* proposed a floorplan design for high-radix clos NoC [35]. The Manhattan distance between routers was studied. None of the above works discussed the problems of crossings. In optical NoC design, the problem of waveguide crossings were considered by many work. Zhang *et al.* proposed a multilayer nanophotonic interconnection network [36]. Kannan *et al.* proposed a hierarchical three-dimensional floorplanning algorithm [37]. They implemented multiple optical layers to decrease the number of crossings.

Different from the above work, our work decreases the number of crossings in the floorplan of the fat-tree based NoC. We study its traversal distance and interconnect length. The floorplan optimization for the fat-tree based NoC, which is the focus of this paper, is rarely studied before. It is hard to compare our optimization method with the related works. Instead, we compare our floorplans with the traditional "H-tree" designs, which is called type H floorplan in our analysis. The optimized floorplan does not change the fat-tree topology and the routing algorithm, and it can be fabricated on a single optical layer. Therefore, it can be applied in any state-of-the-art fat-tree based NoC.

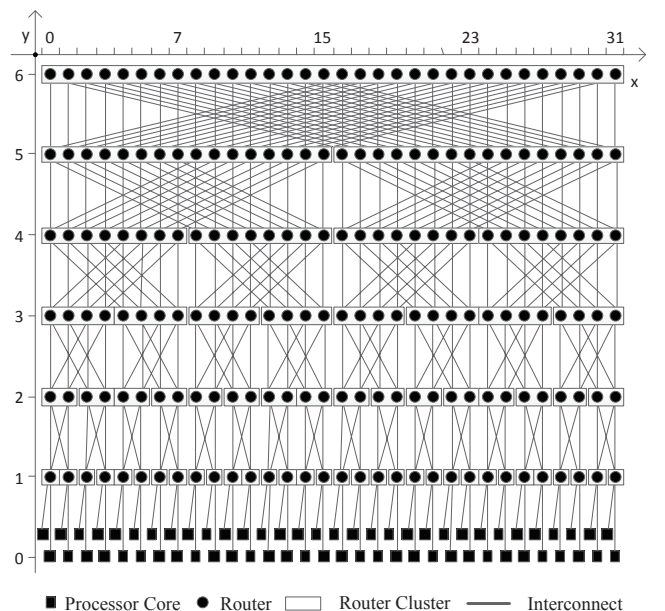


Fig. 1. Network *a*, in fat-tree topology $FT(6)$, there are 32×6 routers and 64 processor cores

3 NETWORK ARCHITECTURE

Routers, processor cores and interconnects are basic components of fat-tree based NoC. The routers can switch signals at a low cost and fast speed. Routers and processor cores are connected by interconnects. Signals can travel freely among all processor cores. The fat-tree topology has a promising future on NoC. In this section, the architecture of NoC that we study in this paper is introduced which includes the topology, algorithm and floorplan. We propose two formats of fat-tree topology including the fat-tree and the general fat-tree. The difference is that fat-tree topology can be derived from the general fat-tree topology.

3.1 Fat-Tree Topology

We first give the definition of fat-tree topology. The traditional floorplan we study in this paper is designed based on this model. There are several fat-tree topologies, which could be derived from the same general fat-tree topology. This fat-tree topology is the most commonly used in floorplan design.

Definition 1. A fat-tree topology $FT(n) = (V, E)$ is a symmetric directed graph with

$$V = \{r(i, j) | i \in [0, 2^n - 1], j \in [0, n]\}$$

$$E = \{e(i, j, k) | i \in [0, 2^n - 1], j \in [1, n], k \in [0, 3]\}$$

Here n is called height. Edges $e(i, j, k)$ are connected as

$$e(i, j, 0) = (v(i, j), v(i - 1, j))$$

$$e(i, j, 1) = (v(i, j), v(i - 1, j'))$$

$$e(i, j, 2) = (v(i - 1, j), v(i, j))$$

$$e(i, j, 3) = (v(i - 1, j'), v(i, j))$$

Here j' is

$$j' = j + (-1)^{\lfloor \frac{j}{2^{i-1}} \rfloor} \cdot 2^{i-1}$$

In $FT(n)$, node $r_{i,0}$ is called processor core, the other nodes are called routers, and the edge is called interconnect.

An example of fat-tree topology $FT(6)$ is depicted in Fig. 1. Each router has four ports, and is capable of switching signals among these four ports. We can observe that there are many crossings in the topology. One goal is to reduce the number of crossings.

3.2 General Fat-Tree Topology

Next, we give the definition of the general fat-tree topology, which has similar definition with the fat-tree topology. The fat-tree topology could be derived from the general fat-tree topology. The difference between fat-tree topology and general fat-tree topology is the routers. We name the routers in the general fat-tree topology as general routers, which also has four ports.

Definition 2. A general fat-tree topology $GFT(n) = (GV, E)$ is a directed graph with

$$GV = \{R(x, y) | x \in [0, 2^{n-y} - 1], y \in [0, n]\}$$

$$E = \{l(i, j, k) | i \in [0, 2^n - 1], j \in [1, n], k \in [0, 3]\}$$

$R(x, y)$ is called general router. It is the set of routers.

$$R(x, y) = \{r(i, j) | i \in [2^y \cdot i, 2^y \cdot (i + 1)], j = y\}$$

Here $r(i, j)$ is the router in $FT(n)$

An example of general fat-tree topology $GFT(4)$ is depicted in Fig. 2. We package routers in $FT(4)$ as one general router in $GFT(4)$. At the same time, interconnect crossings are packaged into the general routers as well.

3.3 General Router

In fat-tree topology, the routers are identical, but in general fat-tree topology, the general routers are different in levels. For example, in $GFT(4)$, there are four levels of general routers, which we name as $R(x, 1)$, $R(x, 2)$, $R(x, 3)$ and $R(x, 4)$. From Fig. 3 we can see that each general router has four ports (indexed from 0 to 3), and each port has the same number of pins.

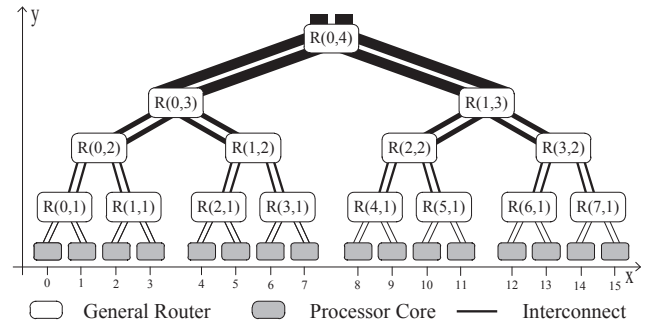


Fig. 2. In general fat-tree topology $GFT(4)$, there are 15 general routers and 16 processor cores

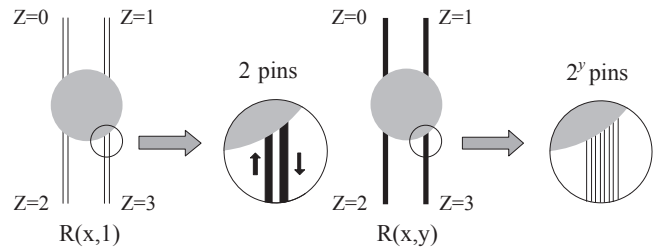


Fig. 3. The general router has four ports, the number of pins in each port depends on the router level

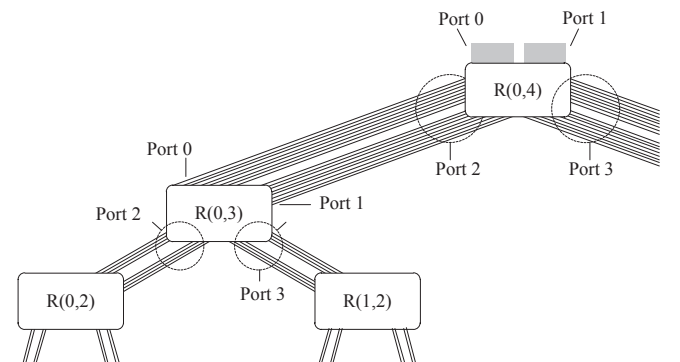


Fig. 4. The general routers $R(0, 4)$, $R(0, 3)$ and their ports, each general router has four ports

Level 1 general router is the same with router in fat-tree topology. Ports are connected by interconnects. Sometimes two ports are connected to one port. In Fig. 4, both port 0 and port 1 of $R(0, 3)$ are connected to port 2 of $R(0, 4)$. Next, we introduce the routing function, which can be defined as follows: given one input port, it generates the output port. Each router is able to transmit signals from the input port to the output port, based on the routing function. There are 10 routing functions in the general router, shown in Fig. 5. One control unit in the general router determines the type of routing functions. The routing functions could be classified into three types: up, down and turn around. For example, signals in port 0 or port 1 can be switched into port 2 or port 3. This is a down process (Function 0, 2, 4, 6). Similarly, in an up process, signals in port 2 or port 3 can be switched into port 0 or port 1 (Function 1, 3, 5, 7). In addition to the up and down process, we have another special function, the turn around process, which means that the general router can switch signals from port 2 to port 3, or the other way around (Function 8, 9).

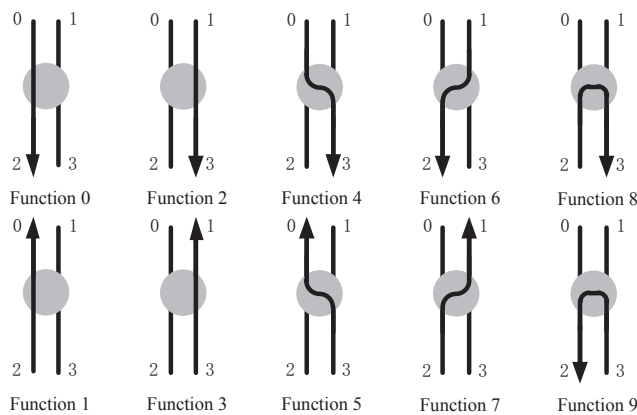


Fig. 5. Each general router has 10 different routing functions, which can be classified into three types

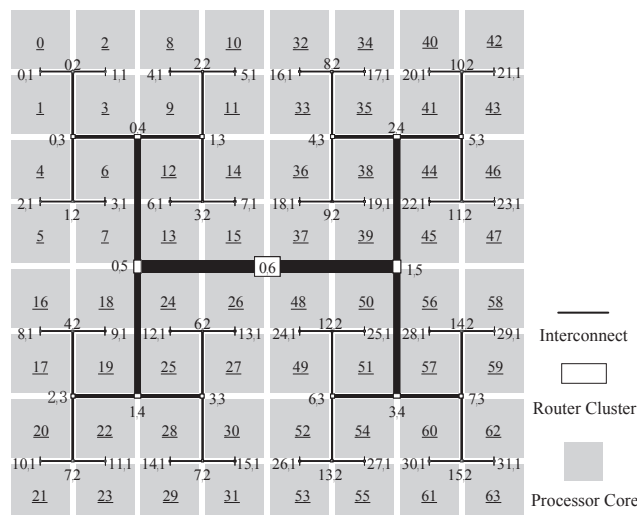


Fig. 6. Traditional floorplan of the fat-tree based network-on-chip using H-tree design (type H)

3.4 Routing Algorithm

In the fat-tree based NoC, the throughput is large, and one processor core can send packets to any of the other one. A packet will experience three stages: the up stage, turn around stage and down stage. For example, in Fig. 2, if the leftmost processor core sends packet to the rightmost processor core, this packet will first go to router $R(0, 1)$, then $R(0, 2)$, $R(0, 3)$, finally it reaches $R(0, 4)$. After the up stage, the packet will turn around at $R(0, 4)$ to general router $R(1, 3)$ and comes into the down stage. Afterwards, it will go through $R(3, 2)$, $R(7, 1)$ and reach the rightmost processor core. In another case, if the first processor core sends messages to the second one. The packet will go up to $R(0, 1)$, turn around at that general router and go down to the second processor core. Because those two processor cores are close to each other, the traversal path is short. In fat-tree based NoC, communications between nearby processor cores will involve less routers, turn around sooner, compared with the communications of faraway processor cores. This algorithm model in general fat-tree topology could be applied to most state-of-the-art fat-tree routing algorithms.

3.5 Floorplan Design

The traditional floorplan of fat-tree based NoC is shown in Fig. 6. It is fabricated on a single optical layer. Processor cores are placed as matrix in electronic layers, and they are assumed to be square in shape. This floorplan looks like the letter "H", which is called "H-tree". We have EO-OE interfaces implemented between processor cores and the network. In fat-tree topology (Fig. 1), several routers are put in one cluster. Those routers are placed in the same position in floorplan design. There is no crossing in Fig. 6, because crossings are already hidden by clusters. In fact, the number of crossings grows quickly if the number of processors cores increases. One crossing will cause some attenuation to optical signals.

Firstly, a large number of crossings will cause high power loss. A typical value of power loss per crossing is 0.12 dB [18], and we also evaluate our model with other values in later section. This property limits the scalability of fat-tree based on-chip networks. Secondly, in the traditional floorplan design, signals go along either vertically or horizontally from one location to the other one on the floorplan. It is obvious that there are some shortcuts between two locations, such as diagonal lines. However, we do not expect new crossings to be generated due to the design of the shortcuts. We have to make tradeoffs in the optimization process. Floorplan plays an important role in designing a highly efficient network-on-chip. The propagation delay is related to the traversal distance, and the power loss is related to the number of crossings. A well-designed floorplan makes balance on the traversal distance and the number of crossings. We give more analysis about this in later section.

4 OPTIMIZATION ON CROSSINGS

We first propose two types of fat-tree topologies. Both of them are derived from the general topology, and we call them network *a* and network *b*. It shows that network *b* has less crossings than network *a*. Next, by optimizing network *b*, we propose network *c*, whose number of crossings is further reduced. We analyze those three networks in terms of average and total number of crossings. It shows that network *c* is the one with the smallest values on both the average and total number of crossings. Hence, the floorplan design of fat-tree based NoC is based on network *c*.

4.1 Optimization Step I

In addition to the fat-tree topology $FT(n)$, there are other topologies that can be derived from the general fat-tree topology $GFT(n)$. First, we introduce the three-general-router structure in Fig. 7. Such a structure covers the entire general topology. Each general router is made up of two parts: *A* and *B*. There are two methods to combine these two parts. In type-*a* method, port 0 of the general router is assembled by port 0 of *A* and port 0 of *B*; port 1 of general router is assembled by port 1 of *A* and port 1 of *B*. Interconnects from those ports will have crossings. Port 2 and port 3 of the general router are designed by similar rules. The number of crossings in type-*b* method is half the value of that in the type-*a* method.

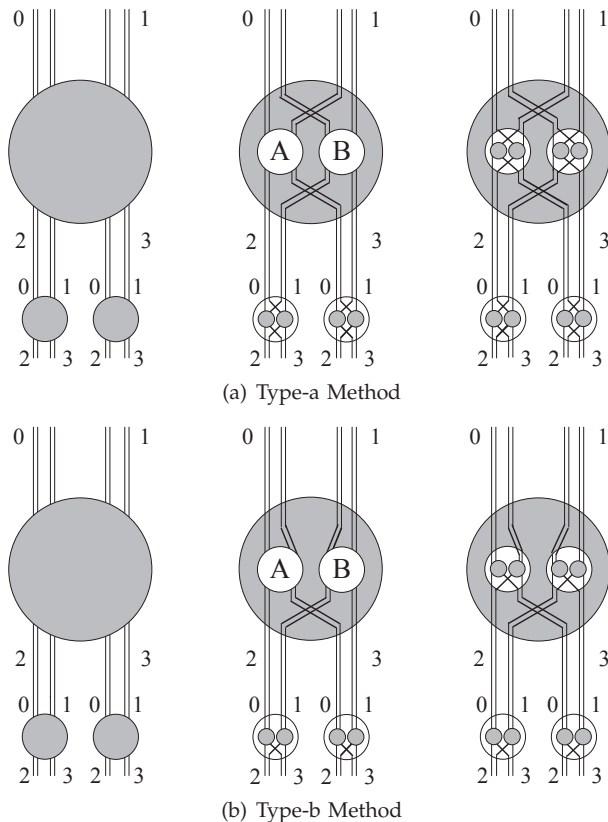


Fig. 7. Two different methods will result in two different networks, but they are equivalent in topology since they are derived from the same general fat-tree topology

In the example of three general routers, the large general router is in level y and the two small general routers are both in level $(y - 1)$. As mentioned earlier, the large general router in level y can be divided into the two general routers: *A* and *B*, which are level $(y - 1)$ general routers. In other words, we can divide the three-general-router structure into four level $(y - 1)$ routers. Similarly, level $(y - 1)$ general router can be divided into two level $(y - 2)$ routers. Therefore, the three-general-router structure can be divided by eight level $(y - 2)$ routers. We can recursively divide the three-general-router structure into level 1 routers. If we recursively use type-*a* method to divide general routers, the resulted network is shown in Fig. 1, which we call network *a*. This network is also the traditional one used in current floorplan designs. On the other hand, if we recursively use type-*b* method to divide general routers, the final network is shown in Fig. 9, which we call network *b*.

Since these two networks are derived from the same general fat-tree topology, they are equivalent in topology. Both of the two networks can hold 2^n processor cores. To show the equivalence, for network *a*, we move the router at position (x, y) to position $(f(x, y), y)$, and maintain the connection between routers and interconnects. The resulted network is the same with network *b*. Function $f(x, y)$ is given as the following:

$$\begin{aligned}
 f(x, 1) &= x \\
 f(x, y) &= f(\lfloor x/2 - \lfloor x/2^{y-1} \rfloor \cdot 2^{y-2}, y - 1) \\
 &\quad + \lfloor x/2^{y-1} \rfloor \cdot (2^{y-1} - 2^{2^{y-3}}) \\
 &\quad - 2^{y-1} \cdot \lfloor x/2 - \lfloor x/2^{y-1} \rfloor \cdot 2^{y-2} \rfloor
 \end{aligned} \tag{1}$$

Both network *a* and *b* are fat-tree topology. They can be directly mapped to physical layouts in two-dimensional space. The layout we mention here is different from floorplans. When the layouts are analyzed, for example, network *a* and network *b*, we are only concerned about the crossings. In floorplan design, on the other hand, the physical length of interconnects and the position of routers are also considered.

To choose the best one from network *a* and network *b*, their numbers of crossings are mathematically calculated. We first evaluate the number of crossings in core-to-core traversal path. There are multiple paths from one processor core to another processor core, and we evaluate the average number of crossings in those paths. In both network *a* and *b*, the patterns between two adjacent rows of routers have similar characteristics. We can use an example to demonstrate those characteristics. The patterns between level y routers and level $(y + 1)$ routers are shown in Fig. 8. There are $2^y \times 2$ routers in those two levels, and we can see that the number of crossings is different in the two examples. We analyze the crossings of each individual interconnect, and define $cross(x, y, z)$ as the number of crossings that a signal will experience when it climbs up from port z of the router $r(x, y)$ to level $(y + 1)$

router. Network a and b have different expressions:

$$\begin{aligned} cross_a(x, y, z) &= 2x - 2xz + (2^y - 1)z \\ cross_b(x, y, z) &= 2x + z \\ x &= 0, 1, 2 \dots, 2^{y-1} - 1, z = 0, 1 \end{aligned} \quad (2)$$

For example, we have the following:

$$cross_a(2, 3, 1) = 7 \quad cross_b(2, 3, 1) = 5 \quad (3)$$

which means that in network a (Fig. 1), the number of crossings from port 1 of router $r(2, 3)$ to level 4 router $r(5, 4)$ is 7, and that value in network b (Fig. 9) is 5.

We calculate the average number of crossings. Define $P(x = i)$ to be the probability that the signal comes out from router $r(i, y)$, and $P(z = k)$ to be the probability that the signal comes out from port k . We assume that this probability is evenly distributed, and they are expressed as:

$$\begin{aligned} P(x = i) &= 1/2^{y-1} \quad i = 0, 1, 2 \dots, 2^{y-1} - 1 \\ P(z = k) &= 1/2 \quad k = 0, 1 \end{aligned} \quad (4)$$

For example, for port 1 of router $r(2, 3)$, we have:

$$P(x = 2) = 1/4 \quad P(z = 1) = 1/2 \quad (5)$$

which means that by 25% chance, signals from level 3 to level 4 starts from router $r(2, 3)$, and by 50% chance, it starts from port 1. It is assumed that signals will climb up to the topmost level routers. For example, in topology $GFT(6)$, the core-to-core traversal path will go through the topmost router in level 6. On that path, there are 11 routers and 12 interconnects.

A new parameter a_j is introduced ($1 \leq j \leq n-1$) to denote the number of waveguides. By definition, interconnects connecting level j and level $(j+1)$ routers have a_j waveguides, respectively. In our analysis, $a_j = 2$ for any j is assumed. It means that interconnects in the whole network have two waveguides. The number of crossings of network a and network b is expressed in Equation (6) and Equation (7).

$$\begin{aligned} average(n) &= 2 \sum_{j=1}^{n-1} (a_j \sum_{i=0}^{2^j-1} (\sum_{k=0}^1 (cross(i, j, k) \\ &\quad P(z = k))P(x = i))) \end{aligned} \quad (6)$$

Assume $a_j = 2$ for $1 \leq j \leq n-1$

$$\begin{aligned} average_a(n) &= 3 \cdot 2^n - 4n - 2 \\ average_b(n) &= 2 \cdot 2^n - 2n - 2 \end{aligned} \quad (7)$$

For example, in topology $GFT(6)$, we have:

$$average_a(6) = 166 \quad average_b(6) = 114 \quad (8)$$

which shows that the average number of crossings on core-to-core traversal path in network b is about 31% less than that of network a .

In addition to the average number of crossings, it is also important to reduce the total number of crossings of $GFT(n)$. This is important because in electronic fat-tree based on-chip networks, less number of vias are implemented. The total number of crossings in net-

work a and network b can be expressed in Equation (9) and Equation (10).

$$total(n) = \sum_{j=1}^{n-1} (a_j^2 \sum_{i=0}^{2^{j-1}-1} \sum_{k=0}^1 (cross(i, j, k) 2^{n-j-1})) \quad (9)$$

Assume $a_j = 2$, for $1 \leq j \leq n-1$

$$\begin{aligned} total_a(n) &= 3 \cdot 2^{2n-1} - n \cdot 2^{n+1} - 2^n \\ total_b(n) &= 2^{2n} - n \cdot 2^n - 2^n \end{aligned} \quad (10)$$

For example, in topology $GFT(6)$, we have:

$$total_a(6) = 5312 \quad total_b(6) = 3712 \quad (11)$$

which shows that the total number of crossings in network b is about 30% less than that of network a . It is clear that network b is better than network a in terms of both the average and total number of crossings. Since each crossing attenuates signal, the power loss can be reduced in network b . As a result, we select network b in optimization step I. Next, the number of crossings in network b can be further reduced by optimization step II, where we find network c . Those networks are all derived from the general fat-tree topology. Therefore, the same routing algorithms can be used in those networks.

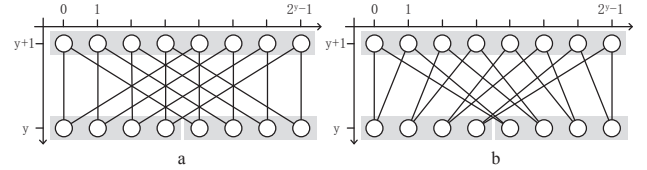


Fig. 8. The comparison of Network a and Network b , both of them have unique patterns

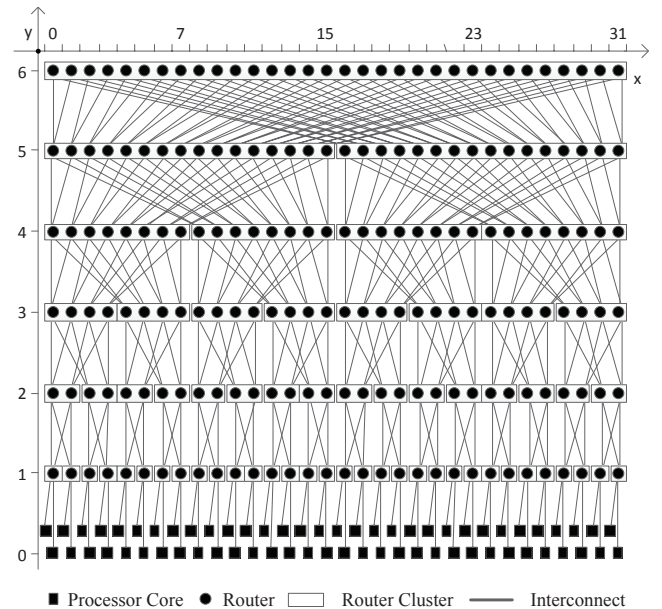


Fig. 9. Network b , this design reduces many unnecessary crossings, compared with Network a

4.2 Optimization Step II

By dividing general routers of the general fat-tree topology into level 1 routers, we find network a and b . Network c is proposed by further optimizing network b . In the general fat-tree topology, it is found that several components in the network can be placed in the same module, shown in Fig. 10. For example, the general router $R(1,3)$ and all its descendant routers and interconnects in the tree are considered as one module. Similarly, the general router $R(2,2)$ and all its descendant components are also in one module. In our definition, a module with general router $R(x,y)$ as a root contains general routers, processor cores and interconnects among them. Any signals coming into or out from this module must go through the general router $R(x,y)$ by 2×2^y interconnects though port 0 and port 1. The module with general router $R(x,y)$ as a root can hold 2^y processor cores. For optimization reasons, we divide each general router in one module into two general routers.

Based on our previous definition, any module can be divided into two modules and one general router. For example, the module with root general router $R(1,3)$ can be divided into the module with general router $R(2,2)$ as a root, the module with general router $R(3,2)$ as a root and the general router $R(1,3)$. After we divide each general router in the general fat-tree topology into two general routers, it is clear to observe some crossings in Fig. 10. In fact, we can assemble the two modules into one module by different methods. There are two methods to assemble two modules, and they are shown in Fig. 11(a). Networks using different assembling methods will have different number of crossings. We assume the general router to be $R(x,y)$. In the first method, there is a crossbar with 2^{2y-2} crossings. In the second method, there are no crossings. Interconnects outside the module are placed upon two opposite directions. It can be seen that the second method can effectively reduce the number of crossings. It is a general way to eliminate crossings, and could be recursively used.

In previous analysis, all the general routers have been divided into two general routers. We can further divide the root general router into four general routers. For the second method mentioned above, there are three cases, shown in Fig. 11(b). In the first case, there are 2^{2y-4} crossings. In the second case, there are 2^{2y-3} crossings, which is two times the value of the first case. The third case could only be used for the top general router in general fat-tree topology. For example, in $GFT(4)$, it could only be used for the general router $R(4)$. Since port 0 and port 1 of general router $R(0,4)$ have no interconnects, there are no crossings in the third case. Similar rules can be found if we further divide the general routers into more general routers. It can be shown that the first case and third case are still good choices that can be used. A new network (Fig. 12) can be found by recursively

using the above methods, and we call it network c . To compare network c with other networks, its number of crossings are mathematically calculated. First of all, The number of crossings on the interconnect between two adjacent rows of routers $cross(x,y,z)$ is:

$$cross_c(x,y,z) = 2\lfloor x/2 \rfloor + z \quad (12)$$

$$x = 0, 1, 2 \dots, 2^{y-2} - 1, z = 0, 1$$

For example, we have the following:

$$cross_c(2,3,1) = 3 \quad (13)$$

which means that in network c (Fig. 12), the number of crossings from port 1 of router $r(2,3)$ to level 4 router $r(5,4)$ is 3. Based on this, we evaluate the average number of crossing on core-to-core traversal path and the total number of crossings in network c . They are expressed in Equation (14).

$$average_c(n) = 2^{n-1} - 2n + 2 \quad (14)$$

$$total_c(n) = 2^{2n-2} - n \cdot 2^n + 2^n$$

For example, in topology $GFT(6)$, we have:

$$average_c(6) = 22 \quad (15)$$

$$total_c(6) = 704$$

which shows that the average number of crossings on core-to-core traversal path in network c is about 81%

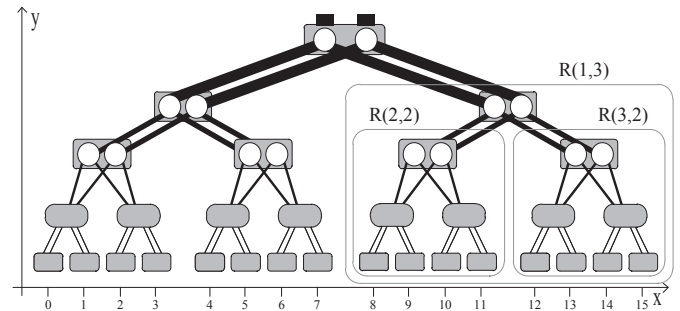
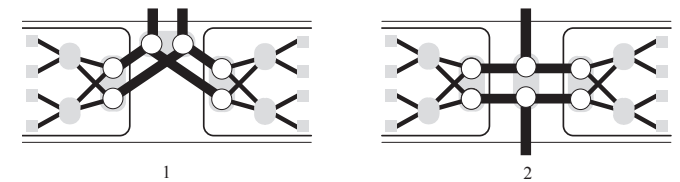
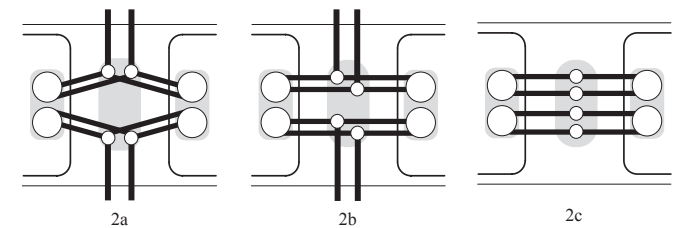


Fig. 10. The general fat-tree topology could be divided into modules in different sizes



(a) Example of Two Methods



(b) Three Cases of the Second Method

Fig. 11. Different methods can be used to assemble two modules and a general router into one module

less than that of network b , and the total number of crossings in network c is also about 81% less than that of network b . It is clear that network c is better than network a and network b in terms of both the average and total number of crossings.

Previously, the number of crossings t is expressed as the function of parameter n . We call this function as $X(n)$, which could be either $average(n)$ or $total(n)$. In our example, $X(n) = average_c(n)$, so we have:

$$t = X(n) = 2^{n-1} - 2n + 2 \quad (16)$$

On the other hand, by finding the inverse function of $X(n)$, we could express the parameter n as the function of number of crossings t . In fat-tree based NoC with n levels of routers, there are 2^n processor cores. The inverse function is:

$$n = X^{-1}(t) = \lfloor \log_2(2t + 4 \lfloor \log_2 t \rfloor + 4) \rfloor \quad (17)$$

For example, if $t = 200$, we have $n = 6$ with 64 processor cores in network a , and $n = 8$ with 256 processor cores in network c . It shows that our optimization on floorplan could increase the network size with the same number of crossings.

In this section, we find three networks: network a , network b and network c , and they can be directly mapped to physical layouts in two-dimensional space. Since they are all derived from the general fat-tree topology, they have the same topology, but different number of crossings. By comparing the first term of

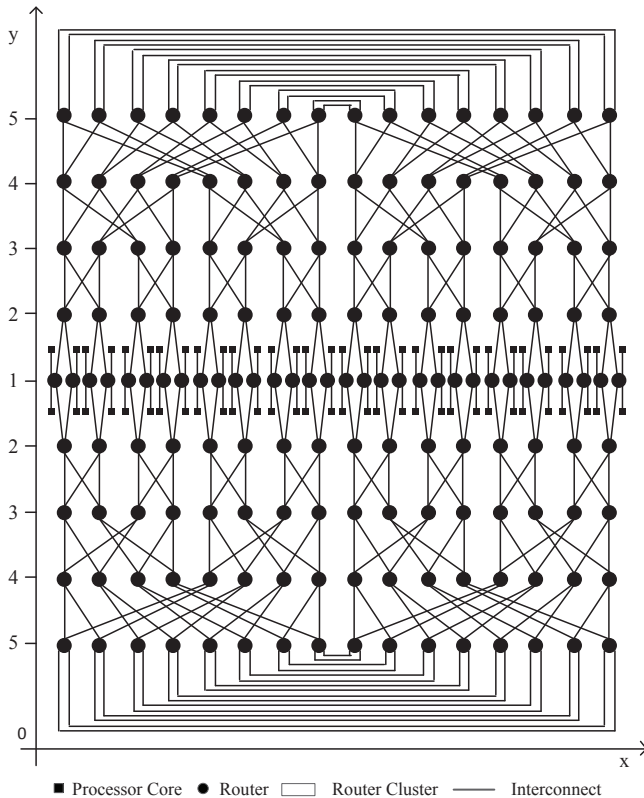


Fig. 12. Network c , it has less number of crossings than both Network a and Network b

Equation (7), Equation (10) and Equation (14), we find that both the average number of crossings and the total number of crossings in network c are about one fourth of the values in network b , and one sixth of the values in network a . It shows that network c has less number of crossings than other networks. If we design floorplans based on one specific network, the number of crossings is unchanged. For example, network c could have many floorplan designs. Those floorplans have the same number of crossings but different interconnect lengths. Network a , on the other hand, is used by the traditional "H-tree" floorplan. All those floorplans are compared in Section 6.

5 FLOORPLAN DESIGN

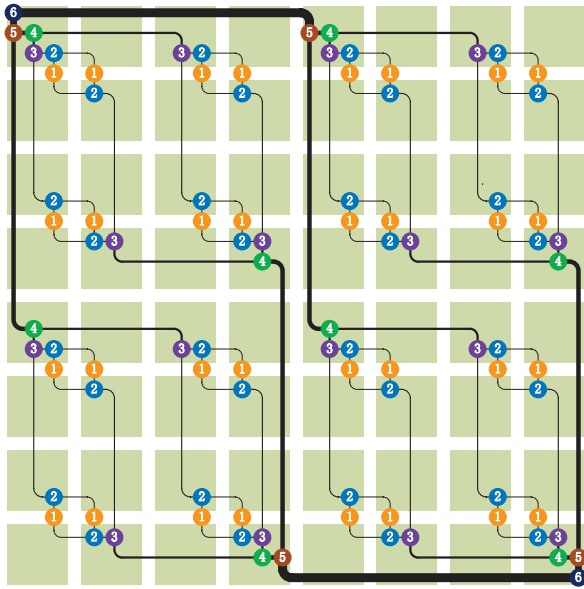
We discuss the floorplan design of fat-tree based optical NoC. The number of crossings has been reduced to the minimum value by previous optimization method. Based on network c , we propose type A and type B floorplan in this section. The traversal distance along the waveguides from core to core also plays an important role in the floorplan design. Both the number of crossings and the core-to-core traversal distance are considered in the floorplan design. Besides, we propose a method to calculate the traversal distance and find the optimal aspect ratio of cores.

5.1 Optimized Floorplan

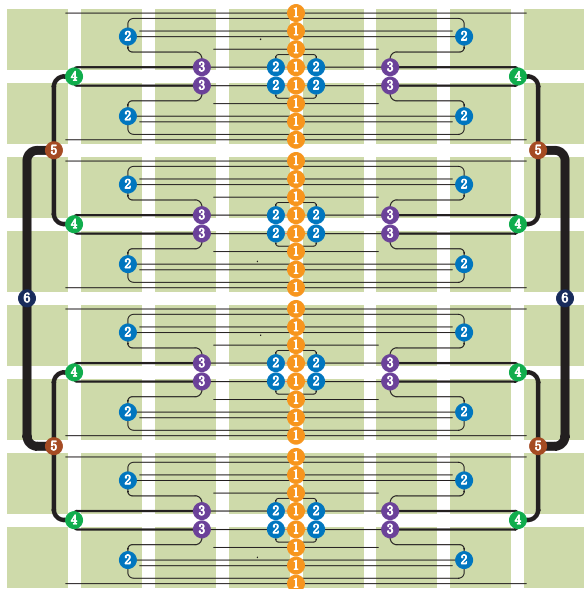
In floorplan design, we assume that a single processor core is rectangular in shape, and a group of cores are fabricated on wafers in matrix. The optical layer is fabricated individually, and placed on top of the wafer. The OE-EO interface is able to convert electronic signal to optical signal and convert optical signal back to electronic signal. They are the main interfaces between the electronic layer and optical layer. Any time one source processor core wants to send a packet to another, signals must go through the interface first, transmitted by optical network and finally come into the destination processor core through the interface. Since there is only one optical layer, the waveguide crossings cannot be avoided. VCSELs and photo detectors are implemented on chip, and they are coupled with waveguides on the optical layer. We assume that circuit switching algorithm is used in our design. An example model has been proposed in [17]. Since the target of this paper is to reduce the number of crossings, packet switching algorithm could also be used. The floorplan design with less number of crossings is also meaningful for electronic NoC because it could decrease the number of vias implemented.

H-tree structure is traditionally used in floorplan design for fat-tree topology. The H-tree is a family of fractal sets and the name "H-tree" is given because it looks like the letter "H". The floorplan of a 64-core network is drawn in Fig. 6. In H-tree floorplan, routers are placed in the junctions of H-tree and interconnects are perpendicular to each other. The topmost general

router, $R(0, 6)$, is placed in the center of the floorplan. The connected two general routers, $R(0, 5)$ and $R(1, 5)$, are placed on its two sides. Locations of other routers and interconnects can be obtained based on this rule. We name H-tree floorplan as type H floorplan. The type A floorplan and type B floorplan are proposed in Fig. 13. Both of them are optimized floorplans with minimum number of crossings. Type A and type B floorplans have different propagation delays due to their different physical traversal distances. According to the analysis in Section 6, type A floorplan has optimal average propagation delay and type B floorplan has optimal maximum propagation delay.



(a) Type A Floorplan



(b) Type B Floorplan

Fig. 13. Type A and B floorplan for 64-core fat-tree NoC, it is assumed that the core is square in shape

5.2 Optimal Aspect Ratio

We propose the method to calculate the core-to-core traversal distance, and find the optimal aspect ratio of cores in a given floorplan. The traversal distance is the summation of interconnect lengths. It is assumed that the core is rectangular in shape. The side lengths are s_1 and s_2 , respectively. We express size vector \mathbf{s} as:

$$\mathbf{s} = (s_1, s_2) \quad (18)$$

The router-to-router interconnect length can be estimated by its ratio to the side length of the core. It is the linear combination of two side lengths. In our method, the lengths of interconnects between level n and level $(n - 1)$ routers are averaged.

$$i_n = r_{1,n}s_1 + r_{2,n}s_2 \quad (19)$$

where $r_{1,n}$ and $r_{2,n}$ are ratios of interconnect length to the lengths of two core sides, respectively. We use ratio matrix \mathbf{R} to express those ratios.

$$\mathbf{R} = \begin{pmatrix} r_{1,1} & r_{1,2} \cdots & r_{1,n} \\ r_{2,1} & r_{2,2} \cdots & r_{2,n} \end{pmatrix} \quad (20)$$

The core-to-core paths can be classified into n categories depending on the number of routers on the path. The traversal distance of path in the m -th category ($1 \leq m \leq n$) is the sum of lengths of interconnects connecting level 1 to level m routers.

$$p_m = 2i_1 + 2i_2 + \cdots + 2i_m \quad (21)$$

where i_n is the length of interconnects between level n and level $(n - 1)$ routers. Signals taking the m -th category path have $2m - 1$ routers and $2m$ interconnects, and it will turn around at level m router. The $n \times n$ constant matrix \mathbf{C} is derived from Equation (19) and (21). We use it to show the included interconnects in the m -th category path.

$$\mathbf{C} = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ & 1 & 1 & \cdots & 1 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & 1 \\ 0 & & & & 1 \end{pmatrix} \quad (22)$$

The core-to-core traversal distance of a given floorplan is the average of paths in different categories. We use different weights to express different number of paths in different categories. The average traversal distance can be expressed as:

$$distance = w_1p_1 + w_2p_2 + \cdots + w_np_n \quad (23)$$

where p_n is the length of traversal path in the n -th category, and w_n is the weight of traversal path with that length. We use weight vector \mathbf{w} to express the weights of different categories of paths.

$$\mathbf{w} = (w_1, w_2, \dots, w_n) \quad (24)$$

In summary, the traversal distance can be calculated by the product of size vector \mathbf{s} , ratio matrix \mathbf{R} , constant

matrix \mathbf{C} and weight vector \mathbf{w} . The result is the linear combination of side lengths of the core. Its minimum value can also be found:

$$distance = 2\mathbf{sRCw}^T = k_1s_1 + k_2s_2 \quad (25)$$

$$\geq 2\sqrt{k_1k_2s_1s_2} \quad (26)$$

In the equation, k_1 and k_2 are the coefficients and are decided by the value of ratio matrix \mathbf{R} and weight vector \mathbf{w} . The coefficient k_1 and k_2 can be derived from Equation (25). Here we give two functions:

$$k_1 = f_1(\mathbf{R}, \mathbf{w}) \quad k_2 = f_2(\mathbf{R}, \mathbf{w}) \quad (27)$$

For given ratio matrix \mathbf{R} and weight vector \mathbf{w} , the minimum value of traversal distance can be found with optimal values of s_1 and s_2 . We assume that the area of one core is $A = s_1s_2$. The optimal aspect ratio of cores can be found based on Equation (28).

$$s_1 = \sqrt{k_2A/k_1} \quad s_2 = \sqrt{k_1A/k_2} \quad (28)$$

We use an example to show our method to calculate the core-to-core traversal distance and find the optimal side lengths and aspect ratio of cores for a floorplan. The type H floorplan is drawn in Fig. 6. Its ratio matrix \mathbf{R} can be directly read from the figure. In this example, interconnects between level n and level $(n-1)$ routers have the same length. In type A floorplan and type B floorplan, we take average values.

$$\mathbf{R} = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 0 & 2 & 0 \end{pmatrix} \quad (29)$$

The weight vector \mathbf{w} is calculated based on probability. Given one source core, we randomly pick up the destination core. It is obvious that by 1/2 chance, signals go to the level n router, and by 1/4 chance, they go to level $(n-1)$ router, and so forth.

$$\mathbf{w} = (1/64, 1/32, 1/16, 1/8, 1/4, 1/2) \quad (30)$$

We assume the area of one core to be $A = 1$. From Equation (27) and (28), we can obtain the optimal side lengths for cores with minimum traversal distance.

$$k_1 = 4.750 \quad k_2 = 3.875 \quad (31)$$

$$s_1 = 0.903 \quad s_2 = 1.107 \quad (32)$$

Finally, based on the above calculation, we can get the average traversal distance of core-to-core paths under optimal aspect ratio of cores. It is expressed as the ratio to the square root of core area.

$$distance = 8.581 \quad (33)$$

In this example, we assume the area of one processor core to be one. Alternatively, we can assume the area of the chip to be one. Therefore, the traversal distance can be expressed as the ratio to the square root of chip area. Based on the above method, the traversal distance of both type A floorplan and type B floorplan can be calculated. The optimal aspect ratios of processor cores can also be found. These three floorplans will be compared and analyzed in next section.

6 ANALYSIS AND COMPARISON

Our proposed type A and type B floorplan and the traditional type H floorplan are compared in this section. We evaluate them based on the core-to-core traversal distance, the router-to-router interconnect length, the number of crossings, the power loss and propagation delay. Analysis shows that both type A and type B floorplans have less number of crossings than type H floorplan. On the other hand, either type A or type B floorplan is better than the other one in some aspects, and we can choose one of these two floorplans.

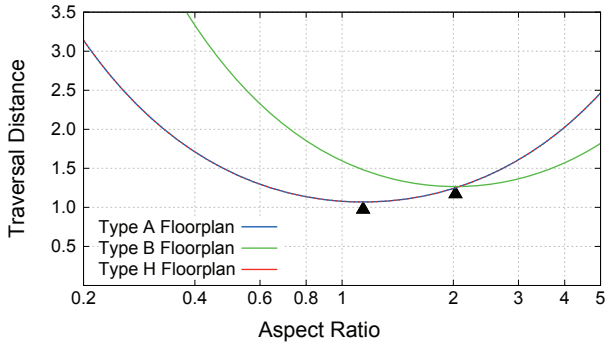
6.1 Core-to-core Traversal Distance

The propagation delay from one core to another core largely depends on the traversal distance between them. We analyze the traversal distance in a 64-core fat-tree network of three floorplans. The cores are placed in an 8×8 matrix and they are in the same size. We assume that the area of the core is given. Hence, the traversal distance depends on the aspect ratio of the core, and it is proportional to the square root of the core area or chip area. In Fig. 14, we show the average and maximum traversal distance as a function of the aspect ratio of core for three floorplans. The maximum traversal distance is the longest in all traversal paths. The traversal distance is expressed by its ratio to the square root of chip area. The minimum traversal distance could be read directly from the chart.

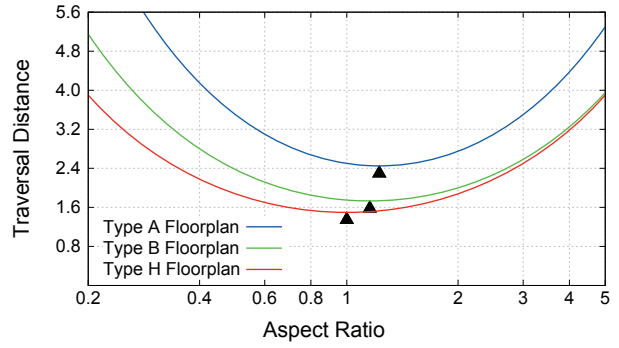
It shows that each floorplan has an optimal aspect ratio of the core. Apart from the maximum traversal distance in type H floorplan, the optimal point does not equal to one, which means that the square-shape processor core is not always the best choice. The average traversal distances of those floorplans are between one to three times of the square root of chip area when the aspect ratio is close to one. For type A floorplan and type H floorplan, the average traversal distance is 0.97 with optimal aspect ratio 1.14. For type B floorplan, the average traversal distance is 1.17 with optimal aspect ratio 2.02. Under those values, the average traversal distance of type B floorplan is about 30% larger than type A floorplan and type H floorplan. Similarly, the maximum traversal distance for type A floorplan is about 30% larger than type B floorplan and type H floorplan. In conclusion, type A floorplan has a smaller average traversal distance, and type B floorplan has a smaller maximum traversal distance.

6.2 Router-to-router Interconnect Length

The interconnect length from one router to another router is a key factor that affects the router-to-router transmission. In packet-switching network, the propagation delay between the two routers equals to the interconnect length divided by the light speed in waveguide. It is assumed that the cores are square in shape. The length of each interconnect can be expressed by its ratio to the side length of the core. We first show the maximum length and the standard deviation of

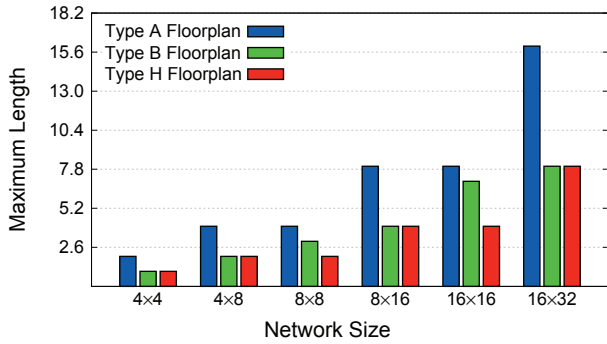


(a) The Average Traversal Distance

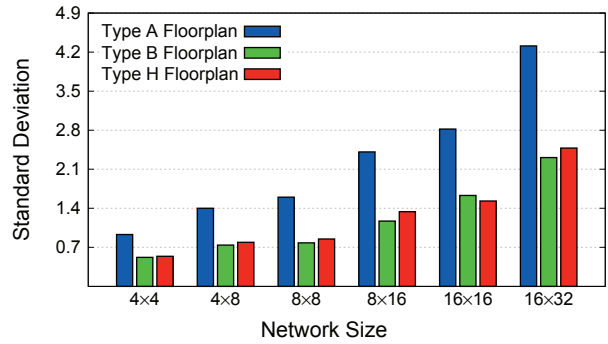


(b) The Maximum Traversal Distance

Fig. 14. The average and maximum traversal distance under various aspect ratios of cores in 64-core fat-tree based network, it is assumed that the chip size equals to one

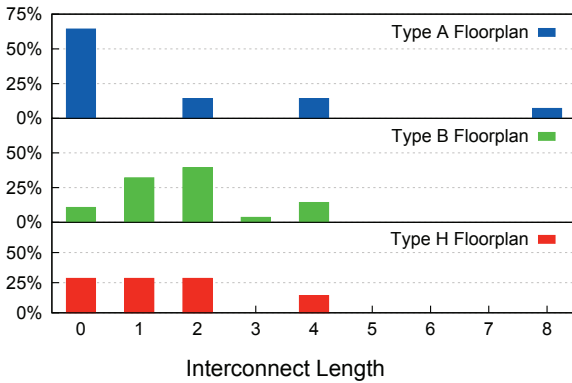


(a) Maximum Length

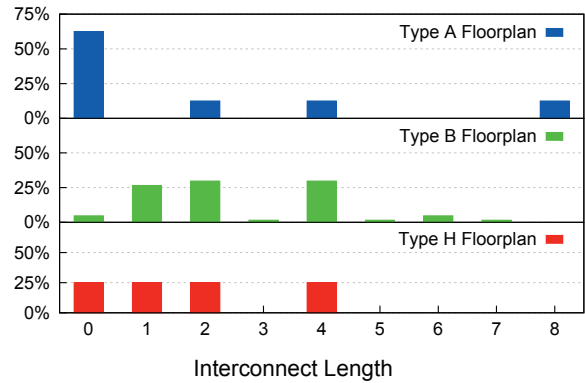


(b) Standard Deviation of Length

Fig. 15. The maximum length and standard deviation of length of interconnects in networks of different sizes, it is assumed that the core is square in shape, the length is expressed by its ratio to the side length of the core



(a) Distribution of 8 × 16 Network



(b) Distribution of 16 × 16 Network

Fig. 16. The interconnect length distribution of networks in two different sizes, it is assumed that the core is square in shape, the length is expressed by its ratio to the side length of the core

interconnects lengths in different size networks, as shown in Fig. 15. It shows that type H floorplan has the smallest value on maximum length under various network sizes. Type A floorplan, on the other hand, has the largest value. Type B and type H has very close values on standard deviation for each network size. The standard deviation of type A floorplan is two times the value of type B and type H floorplan.

Next, we show the percentage of interconnects with different lengths, shown in Fig. 16. In 8x16 network, type A floorplan has a large number of interconnects whose lengths are close to 0 and 8. The distribution of type B floorplan looks like that of type H floorplan. Their interconnect lengths are all smaller than or equal to 4. On the other hand, in 16x16 network, type A floorplan has many interconnects whose lengths equal

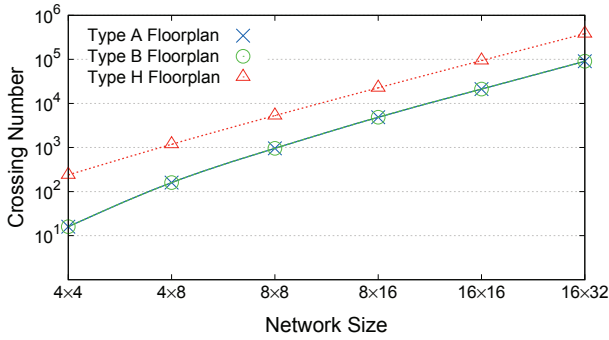


Fig. 17. Total number of crossings in different floorplans under various network sizes, the values of type A floorplan and type B floorplan are exactly the same

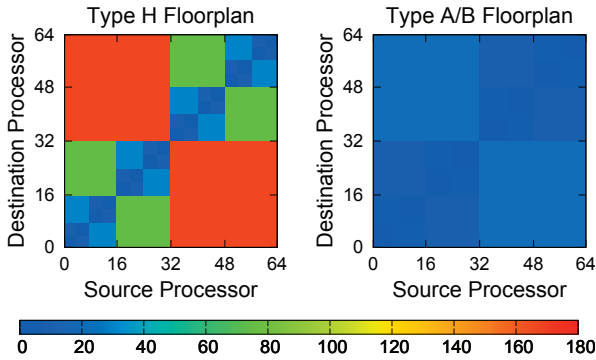


Fig. 18. The number of crossings in core-to-core traversal paths under different floorplans, the value between any two cores is shown in heat map

to 8. Type B floorplan has a smaller number of interconnects whose lengths equal to 7. The interconnect lengths in type H floorplan are smaller or equal to 4. In conclusion, the average interconnect length and the maximum interconnect length of type A and type B floorplan are only marginally increased.

6.3 The Number of Crossings

First of all, we show the total number of crossings in three floorplans, shown in Fig. 17. Both type A floorplan and type B floorplan have the same number of crossings. The total number of crossings in type H floorplan, on the other hand, is about one order of magnitude larger than type A floorplan and type B floorplan under different network scales. The number of crossings in the core-to-core traversal paths is analyzed next. The value between any two processor cores in these three floorplans are shown in Fig. 18. Processor cores are indexed, and each optical path can be distinguished by the combination of a source core index and a destination core index. Processor cores on a chip are indexed in a zigzag fashion, and neighboring cores have close indices. The number of crossings on each path is shown by different colors. Type A floorplan and type B floorplan have the same distribution on number of crossings. In these three

floorplans, paths between neighboring processor cores have less crossings than paths between faraway processor cores. Half paths in the traditional type H floorplan have 166 crossings. On the other hand, all paths in the optimized type A and type B floorplan have less than 22 crossings. The average number of waveguide crossings on traversal path in the optimized floorplan is 87% less than that in the traditional floorplan.

6.4 Power Loss and Propagation Delay

A typical design for optical router, which use micro resonators to switch signals, is assumed [17]. The entire path is setup in advance by a low-overhead network. The VCSELs and photo-detectors are implemented on chip, and they are coupled with waveguides. Besides, we assume the die size to be 10 mm×10 mm. The power loss on traversal path includes crossing loss, micro resonator insertion loss, micro resonator passing loss, waveguide propagation loss, waveguide bending loss and laser/photodetector coupling loss. Each waveguide crossing attenuates signal by 0.12 dB [18]. When signal goes through an on-state micro resonator, it will be attenuated by 0.5 dB [38]. When signal passes by an off-state micro resonator, it will be attenuated by 0.05 dB [38]. Besides, the bending of waveguide will attenuate signal by 0.005 dB every 90 degrees [39]. The propagation loss of waveguide is 0.17 dB every 1 millimeter [39]. On both side of the core-to-core traversal path, on-chip VCSELs and photo-detectors are coupled with waveguide, and each coupler attenuates signal by 1.35 dB [40].

TABLE 1
Average Power Loss on Traversal Path

	Power Loss	Type H	Type A	Type B
Crossing	0.12 dB	134.3	41.6	41.6
MR Insertion	0.5 dB	5.0	5.0	5.0
MR Passing	0.005 dB	12.1	12.1	12.1
Bending	0.005 dB/90°	2175°	2175°	2398°
Propagation	0.17 dB/mm	10.7 mm	10.7 mm	12.7 mm
Coupling	1.35 dB	2	2	2
Total Loss		23.3 dB	12.2 dB	12.5 dB

TABLE 2
Worst Case Power Loss on Traversal Path

	Power Loss	Type H	Type A	Type B
Crossing	0.12 dB	258	60	60
MR Insertion	0.5 dB	11	10	10
MR Passing	0.005 dB	10	10	10
Bending	0.005 dB/90°	2175°	2175°	2398°
Propagation	0.17 dB/mm	15.0 mm	24.5 mm	17.3 mm
Coupling	1.35 dB	2	2	2
Total Loss		41.9 dB	19.2 dB	18.0 dB

TABLE 3
Propagation Delay

	Type H	Type A	Type B
Average Delay (ps)	150	150	178
Maximum Delay (ps)	189	343	242

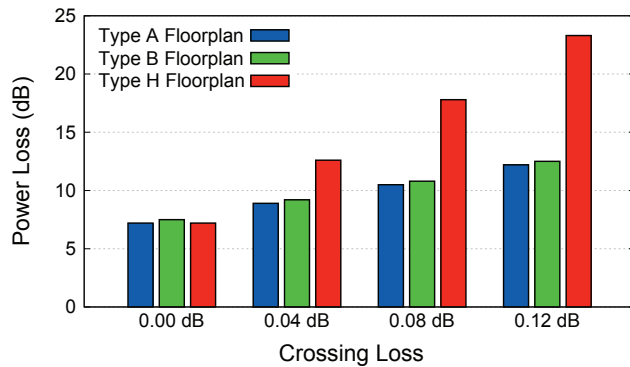


Fig. 19. The average power loss of core-to-core traversal path under different crossing losses

We collect the average number of crossings, micro resonators on the traversal path and traversal distance in Table 1. Similarly, the values of worst case traversal path are collected in Table 2. Both the average and worst case power loss on core-to-core traversal path are calculated based on the values in those two tables. Propagation delay is the traversal distance divided by the light speed in waveguide. The traversal distance is obtained directly from Table 1 and Table 2. The average and maximum propagation delays are shown in Table 3. We also evaluate the power losses under different crossing losses in Fig. 19. Results show that both type A floorplan and type B floorplan can obviously reduce the power loss on traversal path, and their propagation delays are marginally increased, compared with the traditional type H floorplan.

7 CONCLUSIONS

Firstly, we proposed type A and type B floorplans. Compared with the traditionally used type H floorplan, the number of crossings in type A and type B floorplan on core-to-core traversal path is 87 % smaller. The total number of crossings in our floorplans is also one order of magnitude smaller than that of traditional floorplan. On the other hand, our floorplans only marginally increase the traversal distance, compared with the traditional one. Secondly, type A and type B floorplans have different features on traversal distance and interconnect length. Among all the floorplans with optimized number of crossings, type A floorplan has the smallest value on average traversal distance and average interconnect length; type B floorplan has the smallest value on maximum traversal distance and maximum interconnect length. Thirdly, for any floorplan of network-on-chip based on fat-tree, we propose the method to calculate the optimal aspect ratio of processor cores for minimum traversal distance.

ACKNOWLEDGMENTS

The authors are grateful to the reviewers who offer us helpful suggestions to improve this paper. This work is partially supported by RGC of the Hong Kong Special Administrative Region, China.

REFERENCES

- [1] P. Gepner and M. Kowalik, "Multi-core processors: New way to achieve high system performance," in *Parallel Computing in Electrical Engineering*, sept. 2006, pp. 9–13.
- [2] R. Ronen, A. Mendelson, K. Lai, S.-L. Lu, F. Pollack, and J. Shen, "Coming challenges in microarchitecture and architecture," *Proceedings of the IEEE*, vol. 89, no. 3, pp. 325–340, mar 2001.
- [3] J. Henkel, W. Wolf, and S. Chakradhar, "On-chip networks: a scalable, communication-centric embedded system design paradigm," in *VLSI Design, 2004. Proceedings. 17th International Conference on*, 2004, pp. 845–851.
- [4] R. Marculescu and P. Bogdan, "The chip is the network: Toward a science of network-on-chip design," *Foundations and Trends in Electronic Design Automation*, vol. 2, no. 4, pp. 371–461, 2009.
- [5] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Comput. Surv.*, vol. 38, June 2006.
- [6] P. Kapur and K. C. Saraswat, "Optical interconnects for future high performance integrated circuits," *Physica E: Low-dimensional Systems and Nanostructures*, vol. 16, no. 3-4, pp. 620–627, 2003.
- [7] R. Marculescu, U. Ogras, L.-S. Peh, N. Jerger, and Y. Hoskote, "Outstanding research problems in noc design: System, microarchitecture, and circuit perspectives," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 28, no. 1, pp. 3–21, jan. 2009.
- [8] J. Balfour and W. J. Dally, "Design tradeoffs for tiled cmp on-chip networks," in *Annual international conference on Supercomputing*. ACM, 2006, pp. 187–198.
- [9] F. Petrini and M. Vanneschi, "k-ary n-trees: High performance networks for massively parallel architectures," in *In Proceedings of the 11th International Parallel Processing Symposium*, 1997, pp. 87–93.
- [10] S. Ohring, M. Ibel, S. Das, and M. Kumar, "On generalized fat trees," apr 1995, pp. 37–44.
- [11] C. E. Leiserson, "Fat-trees: universal networks for hardware-efficient supercomputing," *IEEE Trans. Comput.*, vol. 34, pp. 892–901, Oct. 1985.
- [12] "Top 500 supercomputer sites," <http://top500.org>.
- [13] C. E. Leiserson, Z. S. Abuhamdeh, D. C. Douglas, C. R. Feynman, M. N. Ganmukhi, J. V. Hill, D. Hillis, B. C. Kuzmaul, M. A. St. Pierre, D. S. Wells, M. C. Wong, S.-W. Yang, and R. Zak, "The network architecture of the connection machine cm-5 (extended abstract)," 1992, pp. 272–285.
- [14] G. M.-S. A. Komornicki and D. Landon, "Roadrunner: Hardware and software overview," IBM, Tech. Rep., 2009.
- [15] P. Pande, C. Grecu, A. Ivanov, and R. Saleh, "Design of a switch for network on chip applications," in *Circuits and Systems*, vol. 5, may 2003.
- [16] A. Andriahantenaina and A. Greiner, "Micro-network for soc: Implementation of a 32-port spin network," *Design, Automation and Test in Europe Conference and Exhibition*, vol. 1, 2003.
- [17] X. Wu, Y. Ye, W. Zhang, W. Liu, M. Nikdast, X. Wang, and J. Xu, "Union: A unified inter/intra-chip optical network for chip multiprocessors," in *Nanoscale Architectures, 2010 IEEE/ACM International Symposium on*, june 2010, pp. 35–40.
- [18] A. Poon, "Cascaded active silicon microresonator array cross-connect circuits for wdm networks-on-chip," *Proceedings of SPIE*, vol. 6898, no. 1, p. 689812, 2008.
- [19] Y. Xie, M. Nikdast, J. Xu, W. Zhang, Q. Li, X. Wu, Y. Ye, X. Wang, and W. Liu, "Crosstalk noise and bit error rate analysis for optical network-on-chip," in *Design Automation Conference*, june 2010, pp. 657–660.
- [20] M. Kreutz, C. Marcon, L. Carro, N. Calazans, and A. Susin, "Energy and latency evaluation of noc topologies," in *Circuits and Systems*, may 2005, pp. 5866–5869 Vol. 6.
- [21] D. Ludovici, F. Gilabert, S. Medardoni, C. Gomez, M. Gomez, P. Lopez, G. Gaydadjiev, and D. Bertozzi, "Assessing fat-tree topologies for regular network-on-chip design under nanoscale technology constraints," in *Design, Automation Test in Europe Conference Exhibitio*, april 2009, pp. 562–565.
- [22] "International technology roadmap for semiconductors for interconnect," 2011.
- [23] R. Beausoleil, P. Kuekes, G. Snider, S.-Y. Wang, and R. Williams, "Nanoelectronic and nanophotonic interconnect," *Proceedings of the IEEE*, vol. 96, no. 2, pp. 230–247, feb. 2008.

- [24] D. Ludovici, F. Gilabert, C. Requena, M. Gmez, P. Lpez, G. Gaydadjiev, and J. Duato, "Butterfly vs. unidirectional fat-trees for networks-on-chip: not a mere permutation of outputs," in *Proc. 3rd Workshop on Interconnection Network Architectures: On-Chip, Multi-Chip*, Paphos, Cyprus, January 2009.
- [25] M. Abd El Ghany, M. El-Moursy, and M. Ismail, "High throughput architecture for high performance noc," in *Circuits and Systems*, may 2009, pp. 2241–2244.
- [26] C. Grecu, P. Pande, A. Ivanov, and R. Saleh, "A scalable communication-centric soc interconnect architecture," in *Quality Electronic Design*, 2004, pp. 343–348.
- [27] H. Hossain, M. Akbar, and M. Islam, "Extended-butterfly fat tree interconnection (efti) architecture for network on chip," in *Communications, Computers and signal Processing*, aug. 2005, pp. 613–616.
- [28] V. Viswanath, "Multi-log processor - towards scalable event-driven multiprocessors," in *Digital System Design, 2004. DSD 2004. Euromicro Symposium on*, aug.-3 sept. 2004, pp. 279–286.
- [29] C. Batten, A. Joshi, V. Stojanovic, and K. Asanovic, "Designing chip-level nanophotonic interconnection networks," *Emerging and Selected Topics in Circuits and Systems, IEEE Journal on*, vol. 2, no. 2, pp. 137–153, june 2012.
- [30] U. Ogras, P. Bogdan, and R. Marculescu, "An analytical approach for network-on-chip performance analysis," *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol. 29, no. 12, pp. 2001–2013, dec. 2010.
- [31] H. G. Lee, N. Chang, U. Y. Ogras, and R. Marculescu, "On-chip communication architecture exploration: A quantitative evaluation of point-to-point, bus, and network-on-chip approaches," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 12, no. 3, pp. 23:1–23:20, May 2008.
- [32] J. Kim, J. Balfour, and W. Dally, "Flattened butterfly topology for on-chip networks," in *Microarchitecture*, dec. 2007, pp. 172–182.
- [33] H. Matsutani, M. Koibuchi, Y. Yamada, D. Hsu, and H. Amano, "Fat h-tree: A cost-efficient tree-based on-chip network," *Transactions on Parallel and Distributed Systems*, vol. 20, no. 8, pp. 1126–1141, aug. 2009.
- [34] T. Ye and G. De Micheli, "Physical planning for on-chip multiprocessor networks and switch fabrics," in *Application-Specific Systems, Architectures, and Processors*, june 2003, pp. 97–107.
- [35] Y.-H. Kao, N. Alfaraj, M. Yang, and H. Chao, "Design of high-radix clos network-on-chip," in *Networks-on-Chip, 2010 Fourth ACM/IEEE International Symposium on*, may 2010, pp. 181–188.
- [36] X. Zhang and A. Louri, "A multilayer nanophotonic interconnection network for on-chip many-core communications," in *Design Automation Conference*, june 2010, pp. 156–161.
- [37] S. Kannan and G. S. Rose, "A hierarchical 3-d floorplanning algorithm for many-core cmp networks," in *Circuits and Systems*, may 2011, pp. 1211–1214.
- [38] S. Xiao, M. H. Khan, H. Shen, and M. Qi, "Multiple-channel silicon micro-resonator based filters for wdm applications," 2007, pp. Opt. Express 15 7489–7498.
- [39] F. Xia, L. Sekaric, and Y. Vlasov, "Ultracompact optical buffers on a silicon chip," *Nat Photon*, vol. 1, no. 1, pp. 65–71, Jan. 2007.
- [40] F. Doany, C. Schow, C. Baks, D. Kuchta, P. Pepeljugoski, L. Schares, R. Budd, F. Libsch, R. Dangel, F. Horst, B. Offrein, and J. Kash, "160 gb/s bidirectional polymer-waveguide board-level optical interconnects using cmos-based transceivers," *Advanced Packaging, IEEE Transactions on*, vol. 32, no. 2, pp. 345–359, may 2009.



Zhehui Wang received B.S. degree in electronic engineering from Fudan University, China in 2010. He is currently working towards the Ph.D. degree at the Department of Electronic and Computer Engineering in Hong Kong University of Science and Technology (HKUST). He served as a technical reviewer of ACM Transaction on Embedded Computer Systems and the 6th IEEE International Conference on Anti-counterfeiting, Security, and Identification. His research interests include embedded system, multiprocessor systems, network-on-chip, and floorplan design for network-on-chip.



Jiang Xu (S02-M07) received Ph.D. degree from Princeton University in 2007. From 2001 to 2002, he worked at Bell Labs, NJ, as a Research Associate. He was a Research Associate at NEC Laboratories America, NJ, from 2003 to 2005. He joined a startup company, Sandbridge Technologies, NY, from 2005 to 2007 and developed as well as implemented two generations of NoC-based ultra-low power multiprocessor systems-on-chip for mobile platforms. In 2007, Dr. Xu joined the

Department of Electronic and Computer Engineering in Hong Kong University of Science and Technology as an Assistant Professor, and established the Mobile Computing System Lab. He currently serves as an Associate Editor of ACM TRANSACTIONS ON EMBEDDED COMPUTING SYSTEMS and IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION SYSTEMS. He is an ACM Distinguished Speaker and a Distinguished Visitor of IEEE Computer Society. He served on the organizing committees and technical program committees of many international conferences. Dr. Xu authored or coauthored more than 60 book chapters and papers in peer-reviewed journals and international conferences. His research areas include network-on-chip, multiprocessor system-on-chip, embedded system, computer architecture, low-power VLSI design, and HW/SW codesign.

Xiaowen Wu Xiaowen Wu received BSc degree in computer science from the Harbin Institute of Technology, China in 2008. He is currently working towards the Ph.D. degree in Electronic and Computer Engineering at the Hong Kong University of Science and Technology. His research interests include embedded systems, multiprocessor systems, and network-on-chip.

Yaoyao Ye Yaoyao Ye received the B.S. degree in Electronic Engineering from University of Science and Technology of China, Hefei, China, in 2008. She is currently a Ph.D. candidate in Electronic and Computer Engineering at Hong Kong University of Science and Technology, Hong Kong. Her research interests include network-on-chip, multiprocessor system-on-chip, and embedded system.

Wei Zhang Wei Zhang received her B.Eng and M.Eng degrees in Electrical Engineering from Harbin Institute of Technology, China in 1999 and 2001, respectively. She received her Ph.D degree from Department of Electrical Engineering in Princeton University at 2009. Dr. Zhang joined Nanyang Technological University, School of Computer Engineering in Jan. 2010. She has published over 20 papers in international conference and journals including a best paper award. Her research interests include embedded systems, system-on-chip, reconfigurable computing, nanotechnology and electronic design automation.

Mahdi Nikdast Mahdi Nikdast received the B.Sc. degree in computer engineering from Islamic Azad University, Esfahan, Iran, in 2009. Since 2009, he has been pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology. He is currently working on SNR analyses for ONoCs in Mobile Computing System Laboratory in HKUST. His research interests include embedded systems, multiprocessor system-on-chip, network-on-chip, and computer architecture.

Xuan Wang Xuan Wang received the B.S. degree in electrical engineering from Shanghai Jiaotong University, Shanghai, China in 2009. From 2009, he has been a Ph.D. student in the department of Electronic and Computer Engineering in Hong Kong University of Science and Technology (HKUST). His research interests include embedded system, multiprocessor system, network-on-chip and fault tolerant design and reliability issues in very deep submicron technologies.

Zhe Wang Zhe Wang received his B.S. degree in Electronic Engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011. He is currently a Ph.D. candidate in Electronic and Computer Engineering at Hong Kong University of Science and Technology, Hong Kong. His research interests include embedded systems, multiprocessor system-on-chip, network-on-chip, hardware/software co-design and design space exploration techniques.