

Floorplanning Exploration and Performance Evaluation of a New Network-on-Chip

Licheng Xue, Weixing Ji, Qi Zuo, Yang Zhang

High Performance Embedded Computation Lab

School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China

{xuelicheng, pass, zql127, young}@bit.edu.cn

Abstract—The Network-on-Chip (NoC) paradigm has emerged as a revolutionary methodology in current System-on-Chips (SoCs) for integrating a large number of processing elements in a single die. It has the advantage of enhanced performance, scalability and modularity, compared with previous bus-based communication architectures. Recently, A new Triplet-based Hierarchical Interconnection Network (THIN) has been proposed. In this paper, we explore the three-dimensional (3D) floorplanning of THIN and present two different floorplanning and routing methods using both the Manhattan routing and the Y-architecture routing architectures. A cycle-accurate simulator is developed based on Noxim NoC simulator and ORION 2.0 energy model. The latency, power consumption and area requirement of both THIN and Mesh are evaluated. The experimental results indicate that the proposed design provides 24.95% reduction in average power consumption and 16.84% improvement in area requirement.

I. INTRODUCTION

Bus schemes are limited for large System-on-Chips (SoCs) because they are inherently non-scalable and produce a huge communication overhead that affect their performance and energy dissipation [1]. Recently, the Network-on-Chip (NoC) paradigm has been proposed as a solution for the interconnect problem, as it offers high bandwidth, low latency, and low power consumption connection between processing elements [2]. Many researchers have optimized the network in various ways, such as developing fast and energy-efficient routers, designing new network topologies, improving the fault-tolerance of the network, and finding better floorplanning designs to improve its performance and decrease its area requirement.

The Triplet-based Hierarchical Interconnection Network (THIN) is a new NoC in chip multiprocessors (CMPs). According to previous studies, THIN has many attractive properties, such as simple topology and computing locality characteristic [3, 4]. However it raises some questions regarding the placement of processing elements, and the routing of wires because of its non-rectangular periphery edge and special memory hierarchy.

The emerging three-dimensional (3D) integrated circuit (IC) [5] and non-Manhattan routing architecture [6–8] provided a new horizon for the floorplanning of the NoC. First, in

3D IC, dies of different types can be stacked with a high bandwidth, low latency, and low power consumption interface, implemented using Through-Silicon Vias (TSVs). Therefore, it improves the performance of the network due to the short global wires. Moreover, cores and high level caches can be distributed to different layers [9], resulting in a more flexible placement of cores and memory blocks in THIN. Second, diagonal wire routing can be easily achieved using a non-Manhattan routing architecture. The similarities of diagonal wires in THIN and Y-architecture make the diagonal wire routing of THIN feasible. The combination of two emerging paradigms, namely, 3D IC and non-Manhattan routing, not only allows for the creation of new structures that enable the floorplanning of THIN, but also provides significant performance enhancements over traditional solutions [10].

This paper investigates the 3D floorplanning of THIN. First, we placed the tile on the core layer, and then routed the wires using both the Manhattan and the diagonal routing methods. The latency, power consumption as well as area requirement for this on-chip interconnect with different floorplanning methods were then evaluated and compared with other common on-chip interconnect architectures, such as 3D Mesh [10]. The simulation results indicate that the THIN outperforms Mesh by 1-5% on average packet latency. In addition, the energy dissipation and area requirement reduction of THIN are 15-25% and 7-20%, demonstrating that THIN is a low latency, power- and area-efficient NoC architecture.

The rest of this paper is organized as follows. The next section provides related work. Section III describes the main architectural features of THIN and its interconnection network strategy. Two floorplanning methods of THIN are presented in Section IV. Section V compares the latency as well as other performance metrics of THIN with 3D Mesh. Conclusions and future work are provided in Section VI.

II. RELATED WORK

Floorplanning for some 2D NoCs has been explored in previous studies [11–14]. In [15] S.Murali proposed an automatic physical planning approach for 2D NoC architecture with Quality of Service (QoS) guarantees. Recently, more studies on 3D NoC have emerged and several works have been done in the area of 3D floorplanning, placing, and routing. 3D floorplanning methods of Butterfly Fat Tree and

This study is supported by the National Natural Science Foundation of China(No.60973010); and the Research Fund for the Doctoral Program of Higher Education of Ministry of Education of China(No.200800071005).

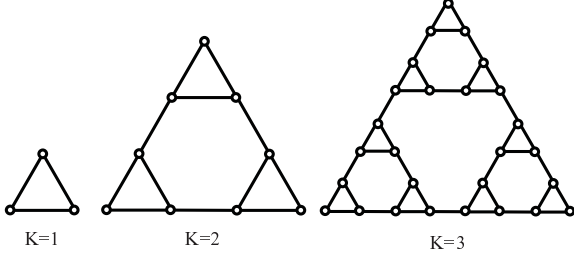


Fig. 1. Interconnection of THIN. The number of processing elements is 3^K .

Fat H-Tree are presented in [16] and [17], in which latency as well as other cost metrics (e.g., energy dissipation and area requirement) has been described. In [10], the authors proposed a multilayer VLSI floorplanning method, evaluated the performance of 3D NoC architectures, and demonstrated their superior functionality in terms of throughput, latency, energy dissipation, and wiring area overhead compared with traditional 2D implementations. The floorplanning of a 3D topology Xbar-connected Network-on-Tiers (XNoTs), which consists of multiple network layers tightly connected via crossbar switches is presented in [18]. Another previous study proposed a synthesis approach for application-specific 3D NoCs [19]. There are also some thermal-aware mapping and placement algorithms for 3D NoCs [20, 21]. Most of recent studies, however, are based on regular networks using Manhattan routing architecture.

Although some Non-Manhattan routing architectures, such as the X-architecture [6] and the Y-architecture [7, 8], have been proposed at the beginning of the 21st century, researchers did not take notice and adopt these routing architectures to implement their topologies. Almost all the discussions on the placement and routing of NoCs are presented using Manhattan routing architecture. In this paper, we used both the Manhattan and Non-Manhattan architectures to achieve the 3D placement and routing of THIN.

III. INTERCONNECTION OF THIN

THIN is a new NoC for chip multiprocessors consisting of a 2D grid of small processing elements, each physically connected to its three neighbors [3]. THIN is a hierarchical network with a number of processing elements increasing by the power of three at each stage. If the value of K represents the levels of hierarchy, then $K=0$ represents a single node. The interconnection strategy for different levels of THIN is shown in Fig. 1. Distributed Deterministic Routing Algorithm (DDRA) has been introduced as a routing algorithm for THIN in [22].

In this paper, we investigate the 3D floorplanning methods of THIN. One advantage of 3D THIN is that the crossbars in THIN have one less port than the widely accepted 3D Mesh. As we known, crossbars scale upward very inefficiently. Large crossbars incur significant area and power overhead over the small ones [23]. TABLE I shows the power consumption and area requirement of routers with different port count.

TABLE I
POWER CONSUMPTION AND AREA REQUIREMENT OF ROUTERS WITH DIFFERENT PORT COUNTS.

	4-port	5-port	6-port	7-port
Power (w)	0.116985	0.148950	0.188681	0.225024
Area (μm^2)	73261	157585	219824	292303

The results were taken from ORION 2.0 [24] with 45nm technology. The supply voltage V_{dd} is 0.8V, and its clock frequency is 3GHz. The experimental results show that the power and area reduction of the router in THIN (5-port, 1 local port, 1 vertical port, and 3 router connection ports) are 26.67% and 39.49% compared with the router in Mesh, which has one more port (6-port, 1 local port, 1 vertical port, and 4 router connection ports).

IV. FLOORPLANNING EXPLORATION OF THIN

Mesh is widely used in parallel computing platform for its simple connection and its ease of placement and routing. Researchers have reached an agreement on the placing and routing method for Mesh. However, other NoCs (e.g., BFT, Torus, Spidergon, and THIN) are not widely accepted partly because the floorplanning of these NoCs is more difficult than Mesh. THIN is physically difficult to implement because of its triangle periphery edge and diagonal wires connecting each router. In this section, we used both Manhattan routing and Y-architecture routing architecture to achieve the placement of the processing elements and the routing of wires in THIN. In order to decrease latency, energy dissipation and area requirement, 3D stacking technology was adopted. Our design is explored based on placing processing elements in the same layer which was closest to the heat sink and leaving cache layers in all remaining layers for the consideration of heat dissipation. We integrated multiple layers by connecting them with a dynamic, time-division multiple-access bus spanning the entire vertical distance of the chip [9]. In this paper, we only presented the floorplanning of the core layer in THIN, since the floorplanning of cache layers are similar to the core layer. We chose 9-core THIN as an example, as both Mesh and THIN are the same sizes and have nine processing elements.

When researchers discuss the routing algorithms for Mesh and THIN, they assume that the cost, including the delay and power consumption of each link, is equal. This equality, however, is established based on the placing and wire routing policies. In this section, we placed the router in THIN in order to make all the links equal first. After the discussion of the equality of wires, we also present a policy which utilizes wires of unequal length to connect the routers.

A. Method 1: Design using Y-architecture

Although THIN is an interconnection network and Y-architecture is a routing architecture, they have some similarities. In THIN, three processing elements make an equilateral triangle, and three equilateral triangles make a larger equilateral triangle. All the links are equal in length, and the angle between two links is either 60° or 120° . The Y-architecture also

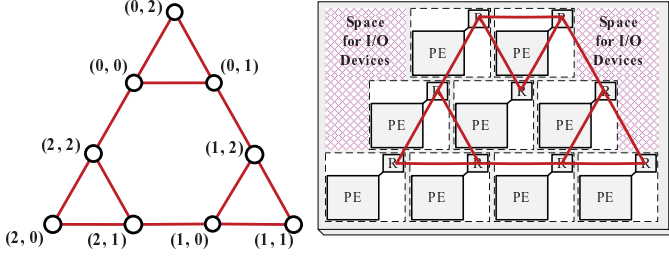


Fig. 2. Y-architecture floorplanning of THIN with equal wires.

has a similar characteristic in that its consecutive orientations are separated by a fixed angle of 60° . The angle between two links is the multiple of 60° . For this reason, the floorplanning of THIN can be easily achieved using Y-architecture.

The floorplanning of tiles and the routing of wires in the core layer are illustrated in Fig. 2. As can be seen, each processing element is placed as a dedicated hard block tile. All the tiles are connected by the wires that are easy to route in Y-architecture. In order to utilize the precious area on chip, the top triangle was rotated 180° and connected to the other two triangles.

However, there are also two differences between THIN and Y-architecture. First, the floorplanning of tiles and routers leaves some space that is not occupied by the processing elements or wires, in this manner, it seems that we waste the precious area on chip. Yet, except for the processing elements, routers and wires, there are also some on-chip I/O devices distributed on the chip, such as the memory controller, the media access controller, physical interface and so on. In our design, we utilized the space on chip as much as possible using these on-chip I/O devices, so as to improve the chip use rate. Second, the nodes that are not located in the periphery of the chip have six degrees of connectivity in Y-architecture. This means that the node is connected with six directly adjacent nodes; but within the interconnection of THIN, each node only has three degrees of connectivity. The degree is the number of ports in the router, apart from the local port, which is used to connect the processing element. One more degree in the node means one more port in the router. TABLE I shows that the power and area reduction of router in THIN (4-port) are 92.35% and 298.99% compared with router in Y-architecture (7-port). If we add the vertical port, the gap will be larger. Therefore, in the implementation of THIN, we simply used the links in Y-architecture and replaced the 7-port router with the 4-port router for low power and low area requirement design.

B. Method 2: Design using Manhattan Architecture

According to different routing architectures, two methods can be used to achieve the triangle interconnection in THIN. Fig. 3 (a) shows an equilateral triangle with a side length of $1.732mm$ (each side of the rectangle tile is $1.5mm$). This type of interconnection can be easily mapped into Y-architecture. Fig. 3 (b) presents another solution, in which the long wire has double the length of the short wires. It can be mapped into the Manhattan routing architecture, which is discussed at

TABLE II
PARAMETERS OF 1 BIT WIRES WITH DIFFERENT LENGTH (5×5).

Length	1.5mm	1.732mm	3mm
Delay (ps)	248	259	310
Link Power (w)	0.092835	0.097011	0.154725

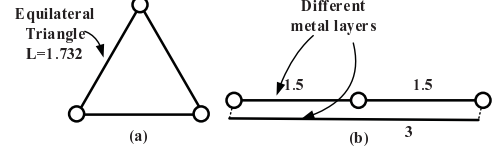


Fig. 3. Methods for connect three nodes in THIN.

the end of this section. If the transmission of the longer wire previously mentioned ($3mm$) can be completed in one cycle, in other words, if we can make all the transmissions in different routing architecture are one cycle transmission, the delay of all wires can be recognized as equal.

For all types of wires, we used HSPICE with 45nm technology low power model from PTM [25]. The supply voltage V_{dd} is $0.8V$, and the operation temperature is supposed to be $70^\circ C$. Repeaters were then inserted in order to accelerate the transmission in long wires. Hence, the transmission in long wires consumes more energy compared with short wires and long wires without repeaters. The resulting delay and power consumption for wires of different lengths are listed in TABLE II. The slowest wire is the $3mm$ wire with a delay of $310ps$. This can sufficiently sustain a $3GHz$ network in such a way that every link requires just one clock cycle to transmit a signal. Therefore, wires of different lengths are allowed as long as the transmission can be completed within one clock cycle.

Subsequently, we performed the second routing method using Manhattan routing architecture which is common and easy to implement. The long wire is adopted based on the previous discussion. The placement and routing of nine processing elements in the core layer are shown in Fig. 4. The placement of tiles in our method is similar to Mesh which is linear and aligned. All the wires are parallel or orthogonal and easy to route. The routing of wires followed the scheme shown in Fig. 3 (b). The tiles of a small triangle in THIN were laid in a linear pattern. The two short wires had a length of $1.5mm$, and the long wire was $3mm$; here, 4 long and 8 short wires were enough to construct the 9-core THIN. The long wires were placed in different metal layers compared with the short wires. Based on the previous discussion, the transmission of all the wires can be completed in a single clock cycle, but the long wires need the addition of repeaters, thereby consuming much more energy.

V. EVALUATION RESULTS AND ANALYSIS

In this section, we present a detailed evaluation of two different floorplanning designs. We compare the network latency, power consumption and area requirement of THIN with 3D Mesh. We do the simulation after the floorplanning because

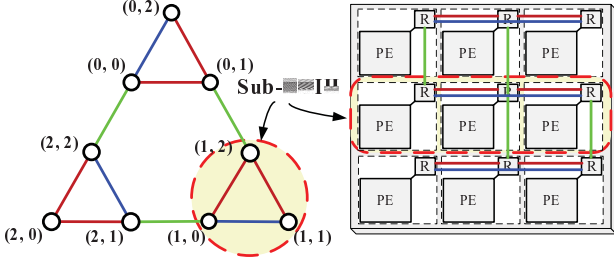


Fig. 4. Manhattan floorplanning of THIN with unequal wires.

TABLE III
CONFIGURATION OF THE SIMULATOR.

Technology	45nm
V_{dd} / Frequency	0.8V / 3GHz
Network	9-core, 1 core and 2 cache layers
VC number / depth	4 / 4 flits
Flit size	128 bits
Routing algorithm	DDRA for THIN, XYZ for Mesh
Traffic pattern	Random, Transpose, Bit Reversal
Warmup / Simulation time	1,000 / 100,000 cycles

the latency and power consumption of link are dependent on the length of the link, and they cannot be accurately estimated without floorplanning information. Next, we will present the detailed evaluation methodology followed by the results.

A. Simulation Infrastructure

To compare the different floorplanning designs, we extended a cycle-accurate 2D NoC simulator Noxim [26] developed in SystemC to a 3D network simulator. It integrated multiple layers of Mesh and THIN networks by connecting them with a dynamic, time-division multiple-access bus spanning the entire vertical distance of the chip. One core layer and two cache layers are used to construct the 3D NoC structure. This simulator allows NoC evaluation in terms of latency and power consumption. The energy parameters of routers and wires were obtained from ORION 2.0 [24] and HSPICE, respectively. ORION 2.0 is an architecture level network energy and area model that can evaluate the power consumption and area requirement of routers.

The parameters used in our simulator are listed in TABLE III. All our experiments were done in 45nm technology. The router had typical components as in a state-of-art NoC router and supported wormhole switching of packets. Each router in THIN and Mesh had 5 and 6 ports, respectively, and each port of the router had 4 virtual channels (VCs). The buffer depth of each VC was 4. We used DDRA routing algorithm in THIN and XYZ routing algorithm in 3D Mesh. Our synthetic workloads consisted of three traffic patterns: uniform random, transpose and bit reversal.

B. Results

1) *Latency*: Fig. 5 plots the average flit latencies for Mesh and THIN using deterministic routing algorithm under the uniform random, transpose and bit reversal traffic patterns. The curves labeled Mesh, THIN_Y, and THIN_M are the

results from standard 3D Mesh, our Method 1 and Method 2, respectively.

In general, the latency increases with the flit injection rate. The latency increases slow at low loads, and soars in heavy injection rate. The results are consistent with our expectations. For uniform random, the latencies of THIN_Y and THIN_M are higher than Mesh by 12.75% and 19.58% on average. Given that the links between routers in THIN are longer than the links in Mesh, the large wire delay increases the latency of packets, even though Mesh has larger and slower routers. However, the other two traffic patterns show the opposite results. There are two reasons. First, THIN has smaller and faster routers. Second, the average hop count is lower between the frequently communicate PE pairs in THIN. For example, the PE0 (0, 0) and PE8 (2, 2) are frequently communicate PEs in transpose and bit reversal. The distance of this PE pair is 4-hop, 1-hop and 1-hop in Mesh, THIN_Y and THIN_M, respectively. The positive impacts conceal the long wire negative impact in these traffic patterns. The average latency improvements of THIN_Y and THIN_M are 15.76% and 10.17%.

2) *Power consumption*: The main power consumption components in NoCs are the wires and the blocks in routers, such as buffers, crossbars and arbitrators. In addition, power due to the clocking of routers is also modeled in our evaluation. To precisely evaluate the relative energy efficiency of the different interconnection networks, we added ORION 2.0 energy model into our simulator and simulated 100,000 cycles for each configuration, collecting energy data.

Fig. 6 summarizes the results with different traffic patterns. Our methods show lower power consumption than 3D Mesh in all traffic patterns. The average power improvements of our methods are 24.95% and 16.36% compared with 3D Mesh, respectively. The Mesh consumes the most energy due to the relatively high connectivity routers at each network hop. Our methods are more energy-efficient as a result of their compact, low-radix routers. The less hop counts also contribute to the improvements. In our method, the power of wires is about 15.37% on average of each tile's total power. Due to the short and low power consumption TSVs in 3D network, the power consumption of links is less than 2D Mesh which consumes about 30% of total power [27].

Until recently, we have assumed some synthetic traffic patterns. In a SoC environment, different functions would be mapped to different parts of the chip, and the traffic patterns would be expected to be localized to different degrees [28]. The advantage of THIN over other topologies, such as Mesh, binary trees and torus, is its efficient exploitation of locality characteristics in complex scientific computations [3, 4]. However, the weak locality synthetic workloads decrease the performance of THIN, and the real workload performance on THIN will be better due to the locality of workloads.

C. Area Requirement

To verify the feasibility of our interconnection network, we considered area requirements of THIN and compared it with

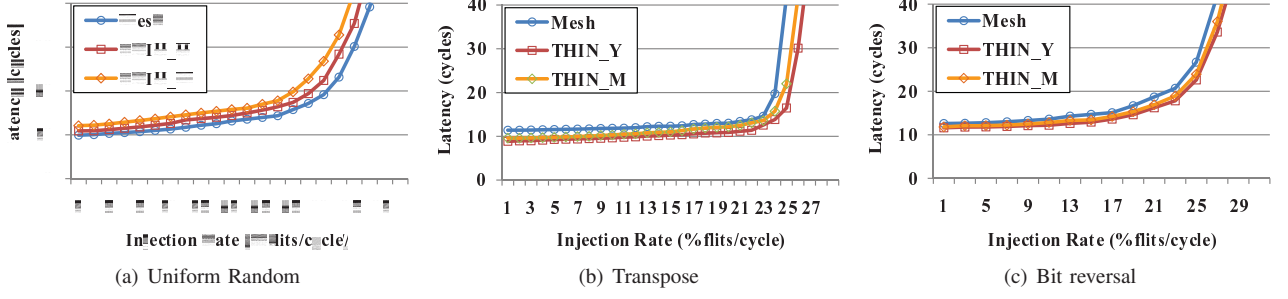


Fig. 5. Average network latency for 9-core Mesh and THIN.

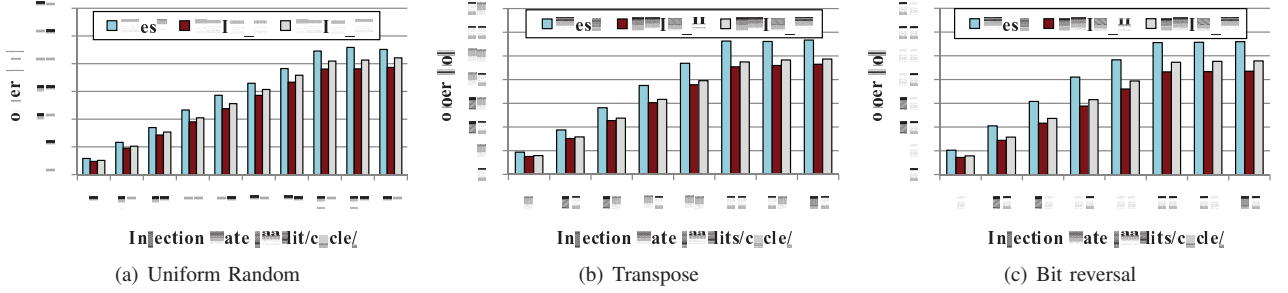


Fig. 6. Average power consumption for 9-core Mesh and THIN.

TABLE IV
AREA REQUIREMENT OF DIFFERENT INTERCONNECTION NETWORKS.

Area (mm^2)	routers	wires	Total
Mesh	1.9782	0.6903	2.6685
THIN_Y	1.4220	0.7970	2.2190
THIN_M	1.4220	1.0355	2.4575

3D Mesh. We developed analytical models to estimate the area of NoC based on the previous discussion on floorplanning. Since the unused space in Method 1 and Method 2 can be occupied by on-chip I/O devices, the area of unused space is not included in the interconnection area.

Generally, the main components concerning area requirement include the wire, the storage buffer and logic to implement routing and flow control. We used ORION 2.0 [24] for developing the area models for wire, buffer and logic.

The area requirement of 3D interconnection network A_{NoC} is determined by the area of routers A_r and links A_l (A_l stands for the area of $1mm$ wire). Therefore, the total interconnection area can be calculated as follows:

$$A_{NoC} = \sum_{j=1}^{N_r} A_r(j) + \sum_{k=1}^{N_l} A_l(k) \quad (1)$$

where N_r is the number of routers, N_l is the length of links in millimeter. TABLE IV summarizes the area requirement comparison of THIN with Mesh where $A_l = 0.03835mm^2$, $A_r = 0.2198mm^2$ for Mesh and $A_r = 0.158mm^2$ for THIN. The area requirement reductions of our designs are 16.84% and 7.91% compared with Mesh. This is probably because the low connectivity routers in THIN occupy smaller area.

D. Scalability

The recursive algorithms of the first floorplanning method is listed in Algorithm 1. The algorithm calls itself in line 6. In order to decrease the area requirement, three sub-networks are rotated (see lines 7-9) and connected by the function connect() which connects the sub-networks through triangle pattern. When Core Count decreases to 1, the single core is returned. For method 2, the algorithm is similar to the first method except that the connection of sub-network is divided into two situations. If K which represents the hierarchy levels of THIN is odd, three sub-networks are arranged and connected horizontally. Otherwise, the sub-networks are arranged and connected vertically.

Algorithm 1 floorplan(CoreCount)

```

1: if CoreCount=1 then
2:   return single_core /*recursive termination*/
3: else
4:   CoreCount  $\leftarrow$  CoreCount/3
5:   /*build three sub-networks*/
6:   top_triangle  $\leftarrow$  left_triangle  $\leftarrow$  right_triangle
        $\leftarrow$  floorplan(CoreCount)
7:   rotate_180(top_triangle) /*rotate the sub networks*/
8:   rotate_120(left_triangle)
9:   rotate_240(right_triangle)
10:  return connect(top_triangle,left_triangle,right_triangle)
11: end if

```

The growth rates of the core counts in THIN and Mesh are different. For THIN, the increase of core count is 3, 9, 27, 81... The increase factor is three. THIN and Mesh only have

the same number of cores when the core count is 9, 81... When the core count rises to 81, the length of the longest wire in the THIN becomes 6mm. The transmission of this long wire cannot be completed within a signal clock cycle. We thus have to insert flip-flops into the wire to make the transmission of flit completed in more than one cycle. Under the configuration, we evaluate that the delay of this type of long wires using HSPICE. Two clock cycles are enough to complete the transmission of long wires. We also evaluated the latency, energy dissipation and area requirement of THIN and 3D Mesh with 81-core. The experimental results with uniform random traffic at 0.05 injection rate are shown in TABLE V. When the core count increases to 81, the latency of THIN is 14.78 cycles which is 11.63% higher than Mesh. However, energy and area reduction of THIN is 22.92% and 20.39%, respectively. The results indicate that our design is feasible for large core count.

TABLE V
COMPARISON OF THIN AND MESH WITH DIFFERENT 81-CORE

	Latency (cycles)	Power (w)	Area (mm ²)
Mesh	13.24	93.64	26.09
THIN_Y	14.78	72.18	20.77
THIN_M	15.34	75.29	22.92

E. Discussion

We arrived at several observations upon analyzing the results of the synthesis workloads. First, the average latency of packets in THIN_Y is a little lower than Mesh (6.26% on average), THIN_M and Mesh have almost the same average latency. Second, the energy reduction of THIN is 15-25% and the area requirement reduction is 7-20% compared with Mesh. All the results indicate that THIN is not only a feasible but also a power- and area-efficient NoC architecture. The Manhattan routing method is thus considered a better approach at present for its high performance and easy implementation. As non-Manhattan routing architecture becomes mature, the non-Manhattan method will improve the performance further.

VI. CONCLUSION AND FUTURE WORK

3D stacking and Non-Manhattan routing technology is used to provide an high performance, energy- and area-efficient interconnection network for CMPs. In this paper, we have explored the 3D floorplanning methods of THIN which is a new NoC. Two possible methods that include a Manhattan design with equal length wires and a Y-architecture design with unequal length wires are investigated. The proposed floorplanning methods achieve the diagonal links of THIN in two different ways. The small routers in THIN contributed to the low power consumption and low area requirement designs.

To ensure a comprehensive evaluation, we utilized a NoC simulator to expose our methods to several synthesis traffic patterns. The experimental results show that THIN is not only a feasible architecture in its physical implementation, but also a power- and area-efficient interconnection architecture.

In order to verify our methods further, we will do more evaluations on THIN using some real workloads in the future.

REFERENCES

- [1] T. Bjerregaard and S. Mahadevan, "A survey of research and practices of network-on-chip," *ACM Computing Surveys*, vol. 38, no. 1, p. 1, 2006.
- [2] L. Benini and G. De Micheli, "Networks on chips: A new SoC paradigm," *Computer*, vol. 35, no. 1, pp. 70–78, 2002.
- [3] F. Shi, B. Qiao, and B. Liu, "A Triplet-based Computer Architecture Supporting Parallel Object Computing," in *ASAP*, 2007, pp. 192–197.
- [4] S. Feng, J. Xinli, and B. Ziru, "Performance of Triplet Based Interconnection Strategy for Multi-Core On-Chip Processors," in *HPCC*, 2009, pp. 163–170.
- [5] S. Das, A. Fan, K. Chen, C. Tan, N. Checka, and R. Reif, "Technology, performance, and computer-aided design of three-dimensional integrated circuits," in *ISPD*, 2004, pp. 108–115.
- [6] T. Ho, C. Chang, Y. Chang, and S. Chen, "Multilevel full-chip routing for the X-based architecture," in *DAC*, 2005, pp. 597–602.
- [7] M. Paluszewski, P. Winter, and M. Zachariasen, "A new paradigm for general architecture routing," in *GLSVLSI*, 2004, pp. 202–207.
- [8] H. Chen, C. Cheng, A. Kahng, I. Mandoiu, Q. Wang, and B. Yao, "The Y-architecture for on-chip interconnect: analysis and methodology," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 24, no. 4, pp. 588–599, 2005.
- [9] F. Li, C. Nicopoulos, T. Richardson, Y. Xie, V. Narayanan, and M. Kandemir, "Design and management of 3D chip multiprocessors using Network-in-Memory," in *ISCA*, 2006, pp. 130–141.
- [10] B. Feero and P. Pande, "Networks-on-chip in a three-dimensional environment: A performance evaluation," *IEEE Transactions on Computers*, vol. 58, no. 1, pp. 32–45, 2009.
- [11] T. Ye and G. De Micheli, "Physical planning for on-chip multiprocessor networks and switch fabrics," in *ASAP*, 2003, pp. 97–107.
- [12] J. Kim, J. Balfour, and W. Dally, "Flattened butterfly topology for on-chip networks," in *MICRO*, 2007, pp. 172–182.
- [13] J. Hu and R. Marculescu, "Exploiting the Routing Flexibility for Energy/Performance-Aware Mapping of Regular NoC Architectures," in *DATE*, 2003, pp. 141–155.
- [14] S. Murali and G. De Micheli, "Bandwidth-constrained mapping of cores onto NoC architectures," in *DATE*, 2004, pp. 896–901.
- [15] S. Murali, L. Benini, and G. De Micheli, "Mapping and physical planning of networks-on-chip architectures with quality-of-service guarantees," in *ASPDAC*, 2005, pp. 27–32.
- [16] H. Matsutani, M. Koibuchi, D. Hsu, and H. Amano, "Three-Dimensional Layout of On-Chip Tree-Based Networks," in *ISPD*, 2008, pp. 281–288.
- [17] A. DeHon, "Compact, multilayer layout for butterfly fat-tree," in *SPAA*, 2000, pp. 206–215.
- [18] H. Matsutani, M. Koibuchi, and H. Amano, "Tightly-Coupled Multi-Layer Topologies for 3-D NoCs," in *ICPP*, 2007, pp. 75–85.
- [19] S. Yan and B. Lin, "Design of application-specific 3D networks-on-chip architectures," in *ICCD*, 2008, pp. 142–149.
- [20] C. Addo-Quaye, "Thermal-aware mapping and placement for 3-D NoC designs," in *IEEE International SOC Conference*, 2005, pp. 25–28.
- [21] M. Pathak and S. Lim, "Thermal-aware Steiner routing for 3D stacked ICs," in *ICCAD*, 2007, pp. 205–211.
- [22] B. Qiao and F. Shi, "A New Hierarchical Interconnection Network for Multi-core Processor," in *ICIEA*, 2007, pp. 246–250.
- [23] J. Kim, C. Nicopoulos, D. Park, R. Das, Y. Xie, N. Vijaykrishnan, M. Yousif, and C. Das, "A Novel Dimensionally-Decomposed Router for On-Chip Communication," in *ISCA*, 2007.
- [24] A. Kahng, B. Li, L. Peh, and K. Samadi, "ORION 2.0: A fast and accurate noc power and area model for early-stage design space exploration," in *DATE*, 2009, pp. 423–428.
- [25] "Predictive technology model," <http://www.eas.asu.edu/~ptm/>.
- [26] F. Fazzino, M. Palesi, and D. Patti, "Noxim," <http://noxim.sourceforge.net/>.
- [27] B. Black, M. Annavaram, N. Brekelbaum, J. DeVale, L. Jiang, G. Loh, D. McCaule, P. Morrow, D. Nelson, D. Pantuso *et al.*, "Die stacking (3d) microarchitecture," in *MICRO*, 2006, pp. 469–479.
- [28] P. Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh, "Effect of traffic localization on energy dissipation in NoC-based interconnect," in *ISCAS*, 2005, pp. 1774–1777.