

Construction of a multidimensional plane network-on-chip architecture based on the hypercube structure

Chang Wu, Yubai Li^{*}, Qicong Peng, Song Chai, Zhongming Yang

DSP Lab, School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

Received 12 March 2008; received in revised form 27 August 2008; accepted 7 October 2008

Abstract

In current network-on-chip (NOC) studies and in practical applications, the mesh structure is the most widely used and deeply researched structure. However, the hypercube structure is more symmetrical and regular than the mesh or torus structures. This paper compares the network characteristics of these three direct topologies and proposes a method to compress the hypercube into a plane structure. This structure, which has the multidimensional property based on the hypercube, is called the multidimensional plane (MDP) NOC. The compression process is divided into two steps: the transformation of router denotations and the connection of channels. Then, SystemC is used to implement the MDP NOC and it is compared with the mesh and torus NOCs in terms of four aspects of performance, including average latency time, normalization throughput, energy consumption and area cost.

© 2008 National Natural Science Foundation of China and Chinese Academy of Sciences. Published by Elsevier Limited and Science in China Press. All rights reserved.

Keywords: NOC; Hypercube; Dimension; SystemC; Simulation

1. Introduction

Bus structures cannot meet the requirements for higher integration and greater intellectual property (IP) re-use on a single chip with increasing complexity [1,2]. Network-on-chip (NOC) has been introduced to address the communication bottleneck posed by the bus structure [3]. In NOC technology, each IP is connected to its own router, and the routers form a communication network-on-chip [4]. Fig. 1 shows the structure of a mesh NOC [5], which is the most widely used structure in NOC applications due to its simple structure and convenient implementation. In the previous NOC studies, the mesh structure was often discussed, despite the fact that torus [6] and hypercube topologies are more regular and symmetrical than the mesh topology. This is because the many loop links existing in these structures make the implementation of the NOC

structure more difficult and complex. For instance, the famous NOC, NOSTRUM [7,8], which was designed by KTH (Royal Institute of Technology Stockholm, Sweden), is still based on a 2D-mesh structure.

However, the mesh NOC is not a high performance NOC when compared with the other NOCs, and it requires three different types of routers, which are vertex routers, boundary routers and center routers. As Fig. 1 shows, there are four vertex routers, eight boundary routers and four center routers. This means that three different router structures should be designed for a single mesh NOC. With increasing IP numbers and NOC scale, the diameter of the mesh NOC quickly increases. As a result, the packets will, on average, take more hops to reach their destination. The equation for the average latency time [9] is

$$\overline{latency} = hop_{ave} \times cycle_{router} \quad (1)$$

where $cycle_{router}$ is the cycle taken for the packet to get through the router and is mostly determined by the router structure. If the $cycle_{router}$ value is the same in different

^{*} Corresponding author. Tel.: +86 28 83207669; fax: +86 28 83201455.
E-mail address: ybli@uestc.edu.cn (Y. Li).

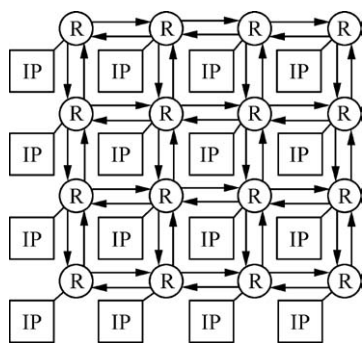


Fig. 1. Mesh topology for NOC.

NOC structures, the latency is completely dominated by the network average hops (the wire delay is ignored). Since the latency is expected to be as small as possible in NOC design, we can only reduce the latency by decreasing the average hops when the router structure has been determined. The other two direct topologies, the torus and the hypercube, have fewer average hops than the mesh. The network properties of mesh, torus and hypercube are compared in Table 1, and the hypercube has the shortest diameter. According to Eq. (1), the hypercube number of average hops is also the smallest among the three topologies. Therefore, NOC based on the hypercube structure has higher network performance.

Since the hypercube is the three-dimensional structure, it should be compressed into a plane structure to adapt to the current NOC applications. With the extension of NOC scale, the dimensions of the hypercube will be larger than three. We should then compress the n -dimensional ($n > 3$) structure into a plane and the corresponding NOC is the multidimensional plane (MDP) NOC.

Taking 4×4 (16 IPs) NOC as an example, the network properties of router degree, channel and diameter are the same in the torus and the hypercube. This means that the three-dimensional hypercube structure can be directly compressed into a plane torus structure. Fig. 2 shows the compression method. If the two surfaces of the bottom cube are interchanged and the hypercube is laid out along with the axis (the dashed line in Fig. 2), the three-dimensional hypercube will be changed into a planar torus. However, as Table 1 shows, there are many differences in structures between the hypercube and the torus with increasing IP number. In the paper, we discuss in detail

the method to compress the hypercube structure into the MDP NOC.

This paper is composed of four parts. Section 2 introduces the details of the method that is used to derive the MDP NOC structures. Through simulation on SystemC, a comparison of the results is given in Section 3 for the performance of the three different NOCs, including average latency time, normalization throughput, power consumption and area cost. Finally, Section 4 draws conclusions from the work in this paper.

2. Compression of MDP NOC

2.1. Transformation of router denotations

Whatever the direct NOC topologies are, their routers can be denoted by some rules when the number of the NOC routers has been determined. According to their positions in the NOC, one of the simplest rules is to denote the routers in terms of the order from small to large. For instance, the routers of 4×4 NOC can be recorded as matrix A , where the numbers represent the denotations of the routers and the relative positions of the numbers correspond to the actual positions of the routers in the practical structures.

$$A = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 \\ 12 & 13 & 14 & 15 \end{pmatrix} \quad E = \begin{pmatrix} 0 & 1 & 3 & 2 \\ 4 & 5 & 7 & 6 \\ 12 & 13 & 15 & 14 \\ 8 & 9 & 11 & 10 \end{pmatrix}$$

$$U = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

From the previous analysis, if matrix A is changed into matrix E , the hypercube is compressed into an MDP structure. Assuming that the transformation matrix is U , then $E = U \cdot A \cdot U$.

When the number of routers increases, their denotations and positions can be gained by a similar method of matrix transformation. For the NOC with $8 \times 8 = 64$ IPs, the original number of its routers can be recoded as matrix B . After the transformation, we get the new denotation matrix, which is recorded as B' .

Table 1
Network properties.

Properties	Topology								
	16 (IPs)			64 (IPs)			$N = 2^n$ (IPs)		
	M	T	H	M	T	H	M	T	H
Degree	2–4	4	4	2–4	4	6	2–4	4	n
Channel	24	32	32	112	128	192	$2(N - N^{0.5})$	$2N$	$n * 2^{n-1}$
Diameter	6	4	4	14	8	6	$2(N^{0.5} - 1)$	$N^{0.5}$	n
Average hops	2.67	2.13	2.13	5.33	4.06	3.04	$2(N^{0.5})/3$	$N^{1.5}/2(N - 1)$	$nN/2(N - 1)$

$$B = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 \\ 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 \\ 32 & 33 & 34 & 35 & 36 & 37 & 38 & 39 \\ 40 & 41 & 42 & 43 & 44 & 45 & 46 & 47 \\ 48 & 49 & 50 & 51 & 52 & 53 & 54 & 55 \\ 56 & 57 & 58 & 59 & 60 & 61 & 62 & 63 \end{pmatrix}$$

Since the MDP NOC is based on the hypercube structure, the new router denotation must satisfy rule 1 below and this rule should also be obeyed by the transformation of router denotations.

Rule 1. Any node in B' has a node address of an n -bit binary number, and has n adjacent nodes, whose node addresses are a Hamming distance of 1 apart from that of the center node.

The detailed steps of the matrix transformation are shown as below:

Step 1. According to the original decimal denotations, use the corresponding binary number to represent every router in matrix B . Such as, 0 is replaced by 000000 and 55 is replaced by 110111.

Step 2. Matrix B is divided into several submatrices.

$$B^{(8 \times 8)} = \begin{pmatrix} B_1^{(4 \times 8)} \\ B_2^{(4 \times 8)} \end{pmatrix}, \quad \text{where } B_1^{(4 \times 8)} = \begin{pmatrix} B_{11}^{(1 \times 4)} & B_{12}^{(1 \times 4)} \\ B_{13}^{(1 \times 4)} & B_{14}^{(1 \times 4)} \\ B_{15}^{(1 \times 4)} & B_{16}^{(1 \times 4)} \\ B_{17}^{(1 \times 4)} & B_{18}^{(1 \times 4)} \end{pmatrix}$$

Step 3. Transform matrix B .

Define R_{ij} as the row transformation matrix and define C_{ij} as the column transformation matrix:

$$R_{ij}^{(4 \times 4)} = \begin{pmatrix} 0 & \cdots & 0 \\ 0 & 1 & 0 \\ 0 & \cdots & 0 \end{pmatrix} \quad C_{ij}^{(8 \times 8)} = \begin{pmatrix} 0 & 0 \\ 0 & I^{(4 \times 4)} \end{pmatrix}$$

In R_{ij} , $0 < i, j < 5$. In this matrix, only one element is “1” (the position is in row i , column j), and the other elements are all “0”. R_{ij} is used to left multiply matrix B_1 or B_2 .

In C_{ij} , $0 < i, j < 3$, and I is the four-order unit matrix and its position is in row i , column j of the partitioned matrix. C_{ij} is used to right multiply matrix B_1 or B_2 .

For example $(R_{ij} \cdot B_1 \cdot C_{ij})$ represents taking the last four columns of the second row from B_1 and putting them to the first four columns of the fourth row in B'_1 . That is,

$$R_{42} \cdot B_1 \cdot C_{21} = R_{42} \cdot \begin{pmatrix} B_{11} & B_{12} \\ B_{13} & B_{14} \\ B_{15} & B_{16} \\ B_{17} & B_{18} \end{pmatrix} \cdot C_{21} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ B_{14} & 0 \end{pmatrix}$$

So, the transforming formula from B_1 to B'_1 is

$$B'_1 = R_{11} \cdot B_1 \cdot C_{11} + R_{21} \cdot B_1 \cdot C_{21} + R_{32} \cdot B_1 \cdot C_{11} + R_{42} \cdot B_1 \cdot C_{21} + R_{13} \cdot B_1 \cdot C_{12} + R_{23} \cdot B_1 \cdot C_{22} + R_{34} \cdot B_1 \cdot C_{12} + R_{44} \cdot B_1 \cdot C_{22}$$

$$\text{Then } B'_1 = \begin{pmatrix} B_{11} & B_{15} \\ B_{12} & B_{16} \\ B_{13} & B_{17} \\ B_{14} & B_{18} \end{pmatrix}$$

In the similar way, B_2 can be transformed to B'_2 , then

$$B' = \begin{pmatrix} B'_1 \\ B'_2 \end{pmatrix} = \begin{pmatrix} I^{(4 \times 4)} \\ 0 \end{pmatrix} \cdot B'_1 + \begin{pmatrix} 0 \\ I^{(4 \times 4)} \end{pmatrix} \cdot B'_2 = \begin{pmatrix} B'_{1L} & B'_{1R} \\ B'_{2L} & B'_{2R} \end{pmatrix}$$

Step 4. Adjust the internal router denotations of B'_{1L} , B'_{1R} , B'_{2L} , B'_{2R} , and obtain the denotation matrix H of the MDP NOC.

$$H = \begin{pmatrix} U \cdot B'_{1L} \cdot U & U \cdot B'_{1R} \cdot U \\ U \cdot B'_{2L} \cdot U & U \cdot B'_{2R} \cdot U \end{pmatrix} = \begin{pmatrix} U & 0 \\ 0 & U \end{pmatrix} \cdot B' \cdot \begin{pmatrix} U & 0 \\ 0 & U \end{pmatrix} = U_2 \cdot B' \cdot U_2$$

From the above-mentioned analysis, it can be noticed that the MDP structure based on the hypercube also has the characteristics that are mentioned below.

H_n (n -dimension MDP NOC) can be composed as a product graph of SH_m and SH_{n-m} ($n > m$), where SH is the subpart of H_n with fewer than n dimensions. This means that H_n can be laid out as a square matrix of nodes with 2^m nodes in every row and $2^{(n-m)}$ nodes in every column, and H_n includes a general torus network with 2^m by $2^{(n-m)}$ nodes as a spanning subgraph in it.

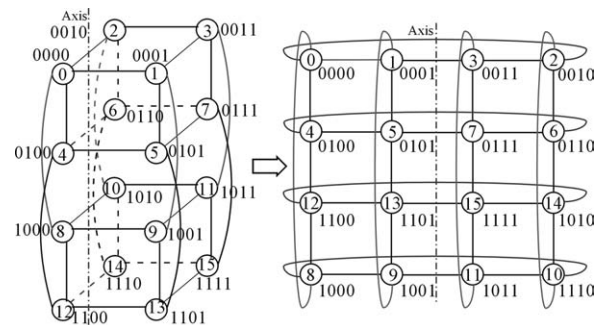
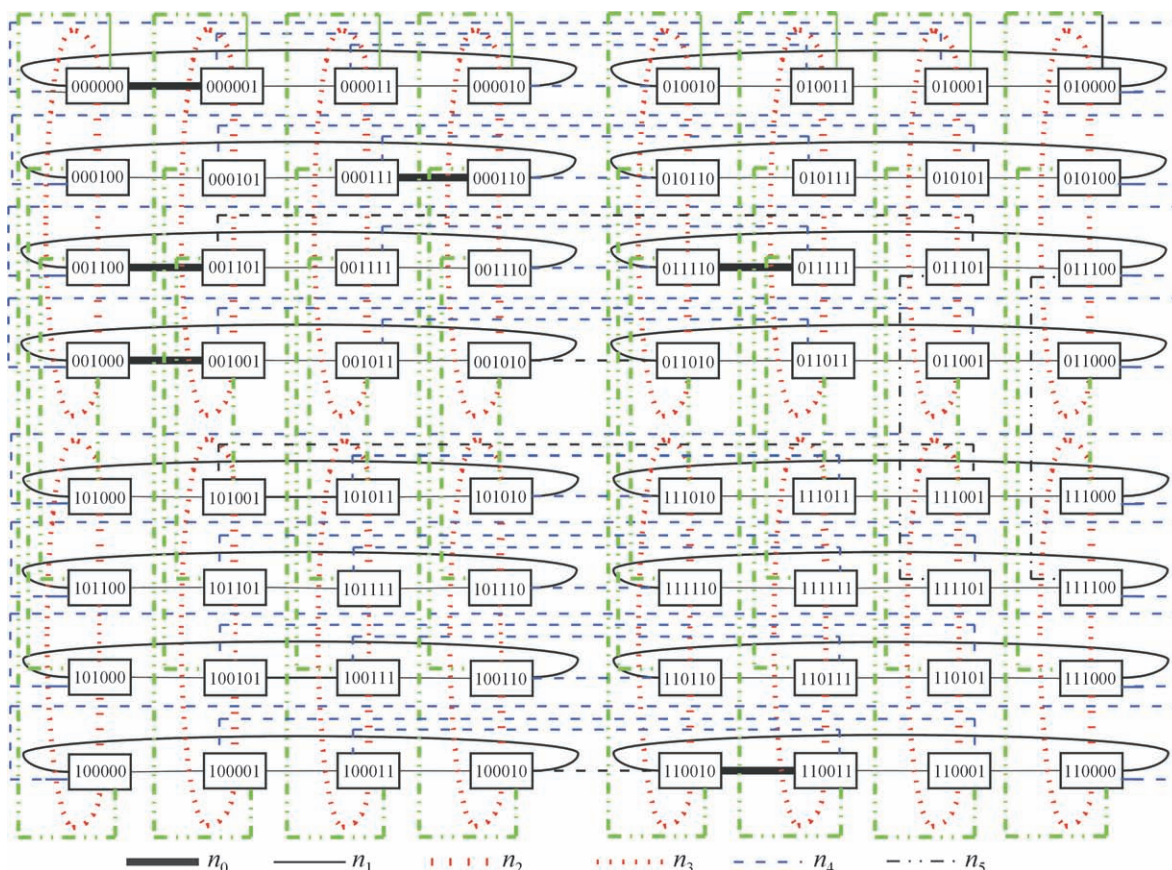


Fig. 2. Hypercube to torus conversion.

Fig. 3. 8×8 (64) IPs MDP NOC.

2.2. Connection of channels

After the denotation transformation, the channels between the routers must be connected in order to get the full structures of the MDP NOC. Since the MDP NOC is changed from the hypercube structure and its routing algorithm is similar to that of the hypercube, the connection of communication channels should satisfy some rules. Firstly, we give some definitions.

Definition 1. H_0 is a single standalone node graph with no edge. H_1 is an undirected graph with two nodes (these are H_0 graphs) connected to each other by an edge; node addresses $(0)_2$ and $(1)_2$ are assigned to these nodes. This means that H_1 can be considered to be composed by 2 H_0 .

Definition 2. H_i ($n \geq i > 1$) is an undirected graph constructed by using two H_{i-1} graphs, whose node addresses are $(0, a_{i-1}, a_{i-2}, \dots, a_1)_2$ and $(1, a_{i-1}, a_{i-2}, \dots, a_1)_2$. If two nodes addresses are the same from a_{i-1} to a_1 , they will be recoded as a pair and connected by channels.

According to H_1 and H_0 , we can get the structure of an arbitrary H_i (the number of routers is 2^i). Therefore, in the 2^n routers MDP NOC, the connection of channels will satisfy the two rules mentioned below.

Rule 2. The channel is undirected and there is only one different bit between the denotations of the two connected routers.

Rule 3. The number of the channels connecting to each router is the same, n . That is, all the degrees of the routers are the same n in the 2^n MDP NOC.

After the denotations transformation and connection of channels, the MDP NOC structure is built and Fig. 3 shows the structure of 8×8 MDP NOC. The six different directions (dimensions) are represented by $n_0 - n_5$. According to the above-mentioned research, we can easily construct the arbitrary MDP NOC structure, where the number of the routers is 2^n .

In addition, there are some long channels in the MDP NOC. At a physical level, this may introduce different link delays, so we split the channels into the same lengths by inserting repeaters [10]. In our study, the extra repeaters are ignored during the research of the routing algorithm because they hardly affect the routing characteristics.

2.3. Routing algorithm

In 2^n IPs MDP NOC, each router connects the other n routers and its degree is n . This is different from the mesh

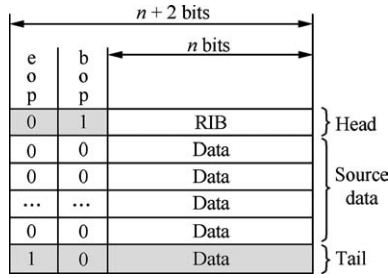


Fig. 4. Form of the data packet.

(Tours) NOC, in which the degree of the router is 2–4 (4), and there are only two dimensions, X and Y , corresponding to four directions $X-$, $X+$, $Y-$, $Y+$. Since the MDP NOC has more dimensions, all the channels are divided into more directions. From Fig. 4, it can be noticed that the channels are divided into six different directions, and each direction corresponds to one bit of the router denotation. Therefore, the routing algorithm of the binary hypercube can be directly used in the MDP NOC. The pseudo code is shown below.

```

for( $i = 0, i < n - 1, i++$ )
{
  if  $dest_i \neq current_i$ 
  then route packet in  $i$  direction;
}
end;
```

This routing algorithm is determined and deadlock free. On this basis, we use SystemC to implement the MDP NOC structure. Then, we give the simulation results and compare the performance of MDP NOC with the mesh NOC and the torus NOC.

3. Performance simulation and analysis

In NOC applications, there are four major guidelines for NOC performance. They are average latency time, normalization throughput, energy (power) consumption and area cost. The average latency time can also be considered as the average end-to-end delay, which is the average time between a packet entering and leaving the network. The normalization throughput is the ratio of successfully transmitted data to injected data. The energy consumption is an important performance measure of the NOC because it is a limitation of chip manufacture. Generally speaking, the energy linearly increases with the increase in the data injection rate. However, when the data injection is beyond the normal load of the NOC, the energy does not increase further, and instead actually decreases. In this paper, the total energy consumption is composed of three parts. They are crossbar, arbiter and buffer energy, where the wire energy consumption is also ignored as being the same as the wire delay. In our simulation, we use the BE mode [11] because the normalization throughput is close to 1 in the GT mode.

The arriving mode of the packets satisfies the Poisson distribution. As shown below, the injection rate, average latency, normalization throughput and energy consumption are defined in Eqs. (2)–(5), respectively.

$$Injection_rate = \frac{number_injection_packets \times L}{cycle_simulation \times number_router} \quad (2)$$

$$Latency = \frac{sum_latency}{number_received_packets} \quad (3)$$

$$Throughput = \frac{number_received_packets}{number_injection_packets} \quad (4)$$

$$E_{total} = E_{crossbar} + E_{arbiter} + E_{buffer} \quad (5)$$

In Eq. (2), the L is the number of flits in a packet.

We use SystemC to implement the NOC structures, including MDP, torus and mesh, which use the routing algorithms of the binary hypercube [12], TXY and XY [13,14], respectively. In NOC applications, the smallest data unit is not a packet but a flit, and several flits compose a packet. The head of the packet contains the routing information, which includes the routing direction, output port or the source and destination addresses. We use routing information bits (RIBs) to store the address information. The packet frame is shown in Fig. 4. The first bit represents the end of packet (eop) and the second bit represents beginning of packet (bop). The statistical information can be stored in the data part of the tail. Some of the sensitive parameters can be recorded in real time during the process of packet transmission. This ensures the availability and authenticity of the statistical results.

In the evaluation of latency, the focus is the saturation point. When the latency reaches twice the minimum delay, this point is considered as being the saturation point of latency and the data injection rate of this point is the saturation injection rate. Before the saturation point, the latency maintains a trend of slow increase. If the injection rate goes beyond the limit of saturation injection rate, the delay will rise dramatically. This means that the NOC reaches the block state and cannot work in the course of nature. Therefore, when the saturation point arrives later, the latency performance of the NOC is better.

Fig. 5 shows the latency in these three different NOCs. From this figure, it can be noticed that the MDP NOC is shown to have the best latency performance among the

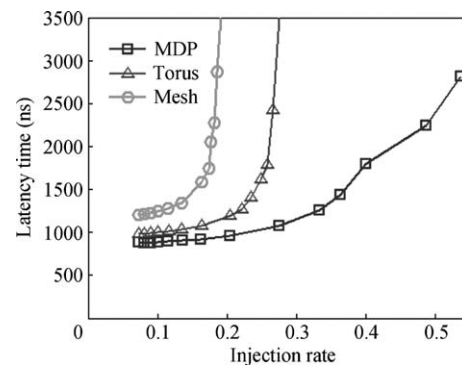


Fig. 5. Average latency time.

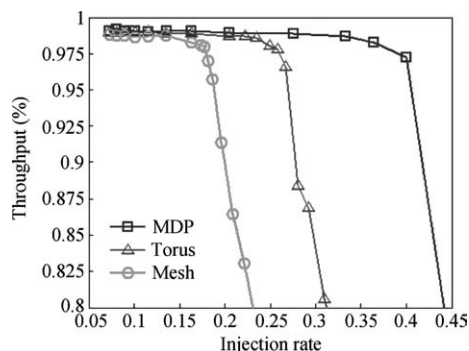


Fig. 6. Normalization throughput.

three structures. This is in accordance with the above-mentioned analysis. In the mesh NOC, the saturation point of latency is located at an injection rate of 0.17. In the torus and MDP NOCs, the saturation points are located at about 0.26 and 0.41, respectively. This means that the saturation injection rate of latency in the MDP NOC is the highest and it can maintain normal running when serious collisions have already happened in the other two NOCs. Therefore, the MDP NOC can afford the highest loads of the three NOCs.

Fig. 6 shows the simulation results of normalization throughput in the three different NOCs. Similar to latency, the saturation point of throughput is defined as the point where the throughput performance drops by half. According to Eq. (4), the theoretical maximum value of normalization throughput is 1. However, the practical throughput is a value less than 1 because the number of data packets entering is always more than that of packets leaving during a period of time. Therefore, the average value of throughput under low injection rate can be regarded as the maximum normalization throughput. For example, the maximum throughput of traffic 1 is around 98.5% in the mesh NOC and the saturation injection rate is 0.17. The rest may be deduced by analogy. From Fig. 8, it can be noticed that the saturation points of throughput are almost the same as those of latency. Therefore, the MDP NOC still has the highest throughput performance among the three NOCs.

We integrate energy evaluation tools called Orion [15] in our SystemC tools to simulate the energy of the NOC.

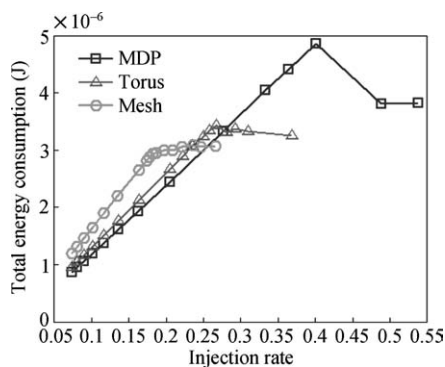


Fig. 8. Energy consumption in three NOCs.

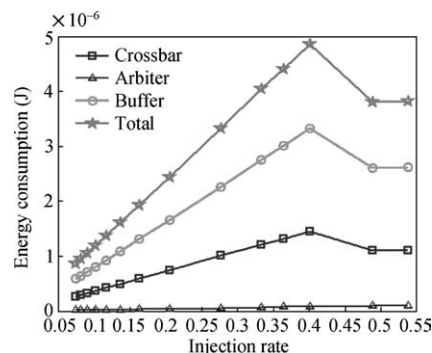


Fig. 7. Energy consumption in MDP NOC.

Fig. 7 shows the energy consumption of the MDP NOC. The energy includes the three parts as shown by Eq. (5). From this figure, the arbiter consumes the least energy of the three parts of the router. With increasing injection rate, the energy consumption will maintain the linear increase until the saturation point arrives. After the saturation point, the energy will present a nonlinear change because the collision induces many blocked packets in the network. The remaining packets cannot enter or leave the NOC and can only be stored in the current routers. This means that the actual number of switching packets has declined. Therefore, the change in the trend of energy curves is irregular. Since it is meaningless to discuss the situation after the saturation point for NOC applications, we focus only on the linear parts of these energy curves.

Fig. 8 shows the curves of the total energy consumption in MDP, torus and mesh NOCs. For the same injection rate, the MDP has the lowest and the mesh has the highest energy consumption. This indicates that the MDP NOC is adapted to the higher integration chip. From this figure, it can be noticed that the MDP NOC can afford heavier loads in larger scale NOC applications, because the linear part of the MDP curve is longer than for the other two structures.

Fig. 9 shows the area cost of the three NOC structures (8 × 8 routers). In the evaluation of the area, the Orion tool ignores the area cost of the wires, and only the areas of the buffer, crossbar and arbiter are taken into account. From this figure, it can be noticed that the MDP NOC has the maximum area, because the router of the MDP NOC is the most complex one of the three NOCs. There-

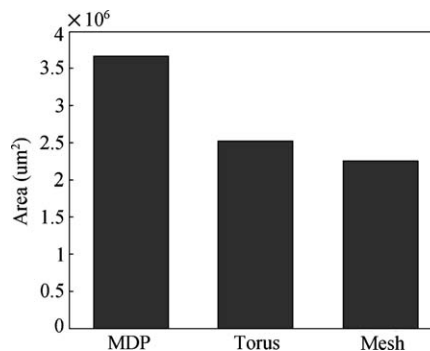


Fig. 9. Area of different NOCs.

fore, the MDP NOC has higher performance but pays the price of greater area cost.

From these simulation results, it is apparent that the MDP NOC has advantages for NOC applications. However, there are still some disadvantages of the MDP NOC. First, the MDP NOC is more complex than the mesh and torus NOCs, so its implementation is more difficult. From current research, the mesh NOC is the most widely researched and used structure in NOC applications because of its relatively simple structure and convenient implementation. Secondly, there are many more communication channels in the MDP NOC. Since our research focuses on the structure and implementation of the MDP NOC, we ignore the effects of the wires in this paper. Thirdly, the MDP NOC is limited by the number of the routers and can only be used with the 2^n routers NOC structure at present. This is the same as the hypercube structure. The MDP NOC is more suitable for a large scale NOC because of its high performance. In future work, we expect to introduce the effects of the wires into our studies.

4. Conclusion

This paper proposes a MDP NOC structure based on the hypercube structure. Through the denotations transformation of the NOC routers, the multidimensional hypercube structure is compressed into a plane. We use matrix transformation to implement the router realignment in 8×8 NOC. According to this method, we can easily deduce the transformation of the 2^n routers NOC. After the channel connection between the routers, the full structure of the MDP NOC is constructed. Then, we use SystemC and the Orion tool to evaluate the four major aspects of the NOC performance, including average latency time, normalization throughput, energy consumption, and area cost. The simulation results show that the MDP NOC has obvious advantages for the first three performance aspects. However, it also required the greatest area of the three structures in $0.18 \mu\text{m}$ technology.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 60575031).

The authors thank Agilent Technologies for their support.

References

- [1] Kumar S, Jantsch A, Soininen JP, et al. A network on chip architecture and design methodology. In: Proceedings of the IEEE computer society annual symposium on VLSI (ISVLSI.02), Pittsburgh, Pennsylvania, USA; 2002. p. 105–12.
- [2] Saleh R, Wilton S, Mirabbasi S, et al. System-on-chip: reuse and integration. *Proc IEEE* 2006;94(6):1050–69.
- [3] Cardoso RS, Kreutz ME, Carro L, et al. Design space exploration on heterogeneous network-on-chip. In: Proceedings of circuits and systems, IEEE international symposium, Kobe, Japan, vol. 1; 23–26 May, 2005. p. 428–31.
- [4] Liu J, Zheng LR, Tenhunen H. Interconnect intellectual property for network-on-chip. *J Syst Archit* 2004;50:65–79.
- [5] Henkel J, Wolf W, Chakradhar S. On-chip networks: a scalable, communication-centric embedded system design paradigm. In: Proceedings of the 17th international conference on VLSI design, Piscataway, NJ, USA; 2004. p. 845–51.
- [6] Pande PP, Grecu C, Jones M, et al. Performance evaluation and design trade-offs for network on chip interconnect architectures. *IEEE Trans Comput* 2005;54(8):1025–40.
- [7] Jantsch A, Tenhunen H, editors. *Networks on chip*. Boston: Kluwer; 2003.
- [8] Penolazzi S, Jantsch A. A high level power model for the Nostrum NoC. In: Proceedings of digital system design: architectures, methods and tools, 9th EUROMICRO conference, Dubrovnik, Croatia; 2006. p. 673–6.
- [9] Dally WJ, Towles B. Principles and practices of interconnection network. Morgan Kaufman Press; 2003.
- [10] Ogras UY, Marculescu R. “It’s a small world after all”: NoC performance optimization via long-range link insertion. *Very Large Scale Integration (VLSI) Syst* 2006;14(7):693–706.
- [11] Rijpkema E, Goossens KGW, Radulescu A, et al. Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip. In: Proceedings of the design, automation and test in Europe conference and exhibition, Munich, Germany, vol. 150, No. 5; 2003. p. 294–302.
- [12] Casavant TL, Tvrdic P, Plasil F. *Parallel computers: theory and practice*. New York: IEEE Computer Society Press; 1996.
- [13] Dally WJ, Towles B. Route packets, not wires: on-chip interconnection networks. In: Proceedings of design automation conference. Las Vegas: ACM Press; 2001. p. 684–9.
- [14] Chiu GM. The odd-even turn model for adaptive routing. *Parallel Distributed Syst* 2000;11(7):729–38.
- [15] Wang HS, Zhu XP, Peh LS, et al. Orion: a power-performance simulator for interconnection networks microarchitecture. In: Proceeding of 35th annual IEEE/ACM international symposium (MICRO-35), Istanbul, Turkey; 2002. p. 294–305.