TNC Transactions on Networks and Communications
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
Vol. 1 Isuue 1

The Communication Performance of Link-Sharing Method of Buffer in NoC Router

The relation between the communication performance and the number of banks

Naohisa Fukase [1], Yasuyuki Miura[1], Shigeyoshi Watanabe[1], M.M. Hafizur Rahman[2]
[1]Graduate School of Technology, Shonan Institute of Technology,
1-1-25, Tsujido Nishikaigan, Fujisawa, Kanagawa, Japan;
[2]Dept. of Computer Science, KICT, International Islamic University, Malaysia;

ABSTRACT

We have proposed a memory sharing method of the wormhole routed network-on-chip architecture. In our method, a memory is shared between multiple physical links by using the multi-port memory. In this paper, we present the pipeline processing method, and evaluate the communication performance in the various situations. The pipeline of the proposed method has two courses of the route 1 and 2. The number of pipeline stages of route2 is 2 stages larger than the traditional router in order to use a shared memory. But delay is concealed if the capacity of a private buffer is enough. It is shown that the required number of memory banks required in multiport memory for 2-dimensional torus and 2-dimensional mesh networks is 8. Our proposed method yields high performance for both torus and mesh networks. Even this high performance is retained when the buffer size and the packet length are same.

Keywords: Router, Interconnection Network, Network-on-Chip (NoC), Multi-Port Memory.

1. INTRODUCTION

Network-on-Chip (NoC) connects hundreds of Intellectual Properties (IPs)/cores, including, programmable processors, co-processors, accelerators, application-specific IPs, peripherals, memories, reconfigurable logic, and even analog blocks. In spite of the many advantages of NoC, area overhead and power consumption still remain the drawback. Therefore, it is necessary to design a high performance router using as minimum hardware resources as possible to minimize the layout area and power consumption.

A single memory is shared by multiple virtual channels for efficient utilization of router buffer is proposed and implemented [1-3]. However, this sharing is taken place in a few virtual channels. For sharing the buffer in more channels or links, we have proposed a buffer sharing method of multiple

N. Fukase, Y Miura, S. Watanabe and M.M. Hafizur Rahman; *The Communication Performance of Link-Sharing Method of Buffer in NoC Router- The relation between the communication performance and the number of banks* in Transactions on Networks and Communications Volume 1, No 1 (2013), PP 01-13
URL: http://scholarpublishing.org/index.php/TNC/article/view/20/TNC-13-1005
Publication Date: December 19th 2013

physical links. Using the proposed method more channels can be shared and the router can utilize buffers more efficiently.

The method of sharing a ring buffer and sharing a multiple buffer are presented in [4] and [5], respectively. However, due to use of large crossbar switch, it is difficult to share large ring buffer. Since wormhole routing is not used in [5], the communication latency becomes prohibitively large because the number of pipeline stages is increased.

In our previous research [6-8], we have proposed the method of sharing a buffer by multiple physical links for effective use of a router buffer. We found that the conventional implementation of sharing technique increase the hardware cost for a large number of physical links. To overcome this problem, we introduce hardware cost reduction method which uses a Multi-bank Multi-port memory [9-10].

In our previous research, we have evaluated the performance of torus network only. It was shown that our proposed method using multi-bank memory has almost same performance with the method using conventional multiport memory when the number of banks is sufficient enough. On the other hand, it is necessary to evaluate the performance when the number of banks is not enough.

The remainder of the paper is organized as follows. In Section II, we briefly describe the conventional method. The proposed method and its hardware cost are discussed in Section III, IV and V, respectively. The communication performance of the proposed method is discussed in Section VI. Finally, in Section VII, we conclude this paper.

2. CONVENTIONAL METHOD

In NoC, a PE consists of one or more processor cores and a router circuit. In router circuit, a crossbar switch is used to connect input links to output in which the communication takes place. A physical link usually has multiple virtual channels [11], and a buffer is integrated to each channel of the input side of the crossbar switch to smooth the flow of packets in communication. Unconstrained use of hardware is strictly prohibited for cost-effective design. Wormhole routing [12] is used for cost-effective design if PE, because it can be implemented by using comparatively a little buffer.

The simple structure of wormhole routers uses a buffer of same capacity installed in each channel [12]. However, the buffer allocated to the channel is not utilized effectively because some channels and buffers remain idle or unutilized. To overcome this problem, sharing a memory by flits between multiple virtual channels of a physical link were proposed and implemented conventionally in [1-3].

By this conventional method, the memory block of a shared memory is assigned dynamically and used when the capacity of the buffer of a channel becomes insufficient. In this method, a

connection between acquired memory blocks is expressed by recording the arrangement of memory to "VC Block Info" of the assigning channel. In this paper, such method is called “Channel Sharing” and the proposed method mentioned in the subsequent section is called “Link Sharing”

3. PROPOSED METHOD

3.1 Outline

Till now the sharing of buffer over a physical link is not used because of increased hardware cost. The link sharing method needs to use a multi-port memory as a shared memory because it responds to the concurrent access from multiple physical links. However, the hardware cost becomes enormous if normal multiport memory is used. It is because the required hardware cost is the square of the number of ports. The structure of the proposed method and the multi-bank multiport memory used in our method are depicted in Figure1 and Figure 2, respectively. As portrayed in Figure 1, each channel has a ‘Private Buffer’ and a shared memory is laid out between input ports in the router. In the proposed method, the ‘Multi-bank Multi-port Memory’ is applied as the shared memory to reduce the hardware cost. As illustrated in Figure 2, the Multi-bank Multi-port Memory has some memory banks which have a few ports, and banks are put between two crossbar switches. In this multi-port memory, it is not necessary to add multiple ports to each memory cell. So it can suppress the increase of hardware cost. However, the Multi-bank Multi-port memory cannot access to addresses in the same bank at the same time.

To solve this problem, we have proposed the ‘By-Block sharing method’. Here, a shared memory is divided into some block and is allocated by every block. By associating each block and bank, the link which accesses to each bank is limited to one. Moreover since the management target becomes a block of the memory, this method can reduce hardware cost. In this paper, the method of controlling a memory by every flit is called ‘By-Flit control’. And the method of controlling by every block is called ‘By-Block control’.

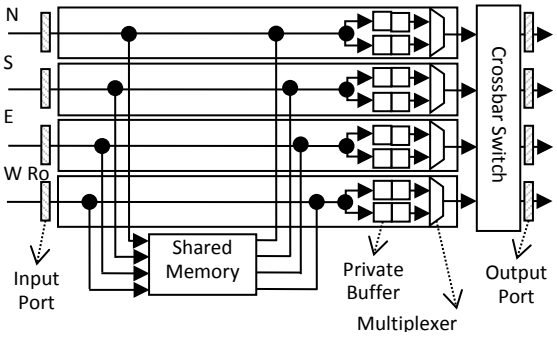

Figure 1: Router Structure of Proposed

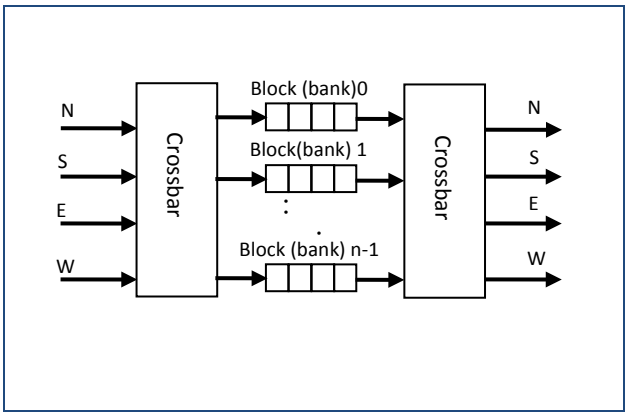

Figure 2 : Multi-bank Multiport Memory

The link sharing method may not allocate a memory to a virtual channel or a physical link due to full of the memory. Thereby, a deadlock [13] may occur. To solve the deadlock problem, a buffer called the 'Private Buffer' of minimum capacity for the communication is laid out to each channel in this proposed method. Even if a shared memory is not allocated, each channel can communicate and can avoid a deadlock.

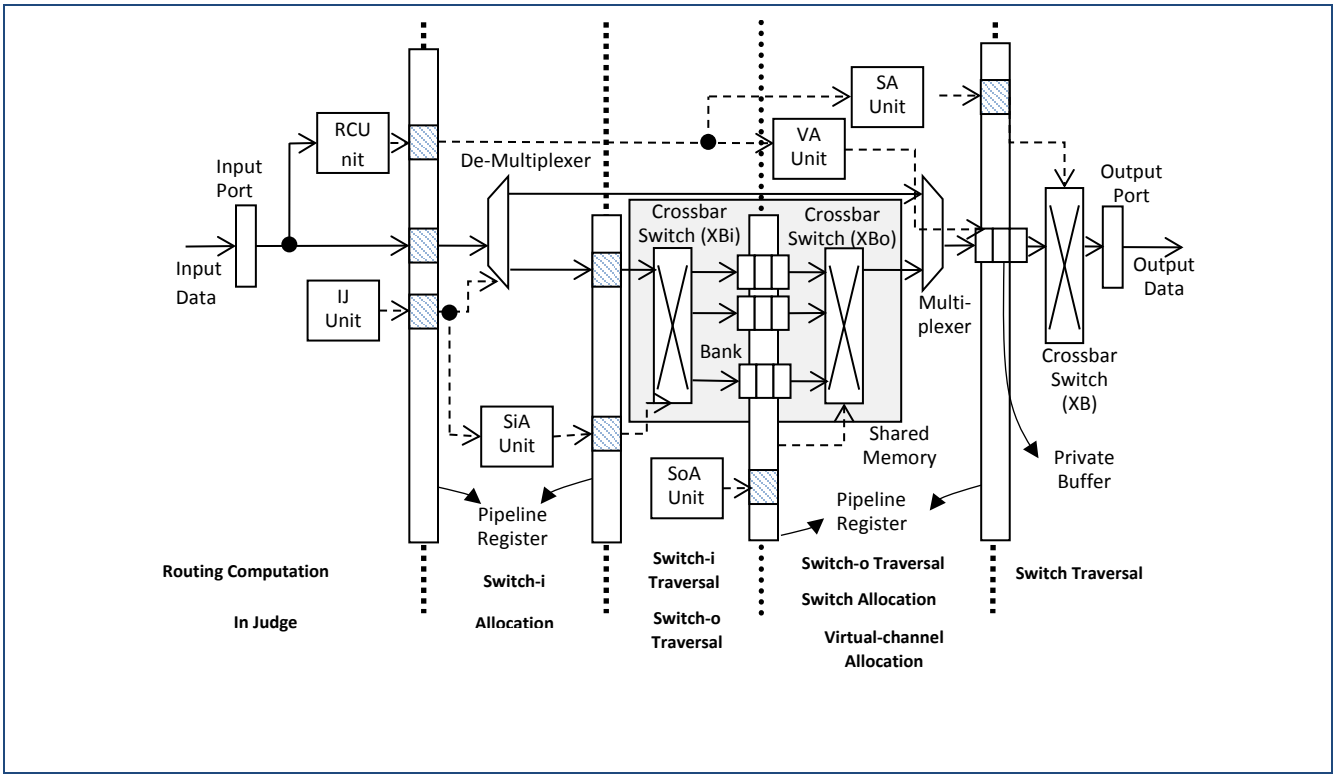

Figure 3: The Block Diagram of Proposed Method

3.2 Hardware Structure

A block diagram including the pipeline structure of the proposed method is portrayed in Figure 3. As depicted in Figure 3, the proposed method has 5 pipeline stages. Each stage is the area surrounded by the dashed line. Each stage is divided by the pipeline register (shown by rectangle in the figure) and buffers such as shared memory and private buffer.

4. PIPELINE STRUCTURE

4.1 Structure of Traditional Router Pipeline

The pipeline structure of the traditional router is shown in Figure 4[14]. As shown in Figure 4, A traditional pipeline performs the following four processes in three steps.

1) Routing Computation (RC) : An output link is determined from the information on header.
2) Virtual Channel Allocation (VA) : The virtual channel to output is assigned.
3) Switch Allocation(SA) : The arbitration and setup of a crossbar switch are performed.
4) Switch Traversal (ST) : Flit passes a crossbar switch.

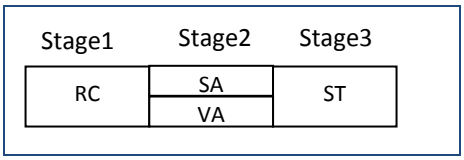

Figure 4 : Pipeline Structure of the Traditional Router

4.2 Pipeline Structure of Proposed Method

In Our method, injected flits pass along the following two routes. When not crowded, it passes along the route 1. When crowded, it passes along the route 2.

route1 : input port⇒private buffer ⇒output port

route2 : input port⇒shared memory⇒private buffer⇒output port

The route 1 is a course immediately sent to Private Buffer after arriving at an input port. The pipeline of the route 1 is shown in Figure 5(a). As shown in Figure 5(a), the route 1 operates on the same three-step pipeline as traditional router. However, In-Judge (IJ) process is executed in stage 1 as shown in Figure 5(a). In IJ process, “Whether the shared memory is used or not”, and “A now block is allocated or not” are determined. Since the output link of a packet is decided regardless of whether a shared memory is used, RC and IJ processes can be processed in parallel.

The route 2 is a course that the packet is sent into the shared memory and then goes to the private buffer after that. The route 2 needs the stage for a setup and traversal of the switch in the input and output port of the Multi-bank Multiport Memory. The pipeline of the route 2 is shown in Figure 5(b). The route 2 needs the following stages:

1. IJ(In-Judge) : “Whether the shared memory is used”, and “A new block is allocated” are determined in IJ process.
2. SiA(Switch-i Allocation) : Set the input crossbar switch of the multi-bank multiport memory.
3. SiT(Switch-i Traversal) : When it succeeds in a SiA step, a packet passes the input crossbar switch of multi-bank multiport memory and it stores to the shared memory.

4. SoA(Switch-o Allocation) : A setup of the output crossbar switch of the multi-bank multiport memory and the block release process are performed simultaneously.
5. SoT(Switch-o Traversal)

When it succeeds in a SoA step, a packet passes the output crossbar switch of multi-bank multiport memory and it stores to the private buffer.

As shown in Figure 5(b), the number of stages of route 2 is 2-stages larger than route 1. But the delay by the pipeline of the route 2 is concealed by following reasons.

- When a network is not crowded and the private buffer is not full, it becomes same stages as the conventional method since it is processed according to a three-stage pipeline (route 1).
- As a network is crowded, the private buffer becomes full and the shared memory will be used. The number of flits in the private buffer increases by blocking the packet by the crossbar switch. If the private buffer is designed to permit one blocking (If the number of flits of the private buffer is two or larger), the pipeline using the shared memory will smoothly flow.

The example of the pipeline of proposed method is shown in Figure 5(c). In Figure 5(c), the 3rd flit goes to the route 2 since top flit was blocked.

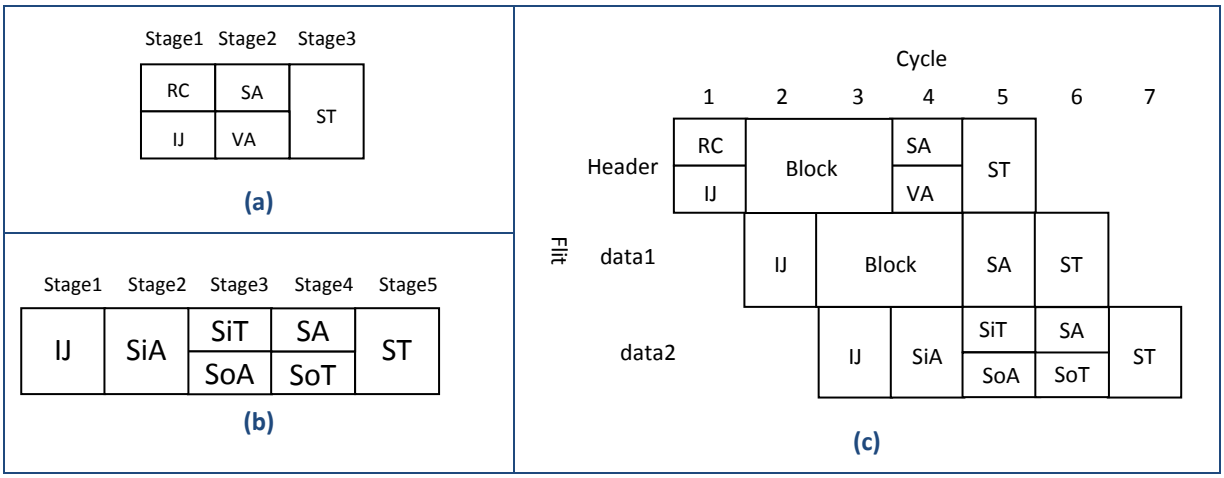

Figure 5 : (a) Route 1 Pipeline, (b) Route 2 Pipeline, (c) The Example of the Proposed Pipeline

5. HARDWARE COST

In this section, the hardware cost to implement the proposed method is estimated. In the conventional method, most of the hardware cost is 'buffer' of the physical link except crossbar switch and control circuit. 'Memory element for control information' is needed for both the traditional method and the proposed method. Memory element includes the buffers for control the shared memory [6-8]. Additional hardware costs for the proposed method are 'logic circuit for block control' and 'surrounded circuits of multiport memory'.

The hardware cost of a physical link can be roughly estimated by estimating the above mentioned elements. In this evaluation, *B*, *C*, *L*, *F*, and *W* are defined as follows:

B: Total number of memory blocks in all links

C: Total number of channels in all links

L: Number of links

F: Number of flits in a block

W: The number of bits per a flit

In this condition, the number of channels per link is C/L, and the number of memory block per link in channel sharing (conventional) method is B/L. Also, in the "By-Flit Sharing", *F* is set as one. The number of transistors for implementation is counted to evaluate the hardware cost. The cost of memory element is assumed as 6, *n*-input NAND (NOR) gate is 2*n*, inverter is 2, the cross point of crossbar switch is assumed to use a tri-state inverter so the number of transistors is assumed as 6.

The implementation cost in terms of the number of transistors of conventional and proposed method is tabulated in Table 1. In the evaluation the total amount of buffer is kept same ($B \times F$=64) and the number of blocks (*B*) are varying. For both the conventional method and by flit and link sharing as shown in Table 1, the value of *F* is equal to 1.

It is shown in Table 1 that the hardware cost of the proposed method decreases with the decrease of the number of blocks (the value of *B* become smaller). The hardware cost can be drastically reduced compared with by-flit implementation (*F=1*). Although the additional logic

Table 1. Implementation Cost of Proposed Method (Transistors)

W	Topology	*L*	*C*	*B*	*F*	Conventional Method	By Flit and Link Sharing	Proposed Method	
								Total	**Improvement rate**
64	Ring	2	4	16	4	30732	154318	**58146**	**1.89203**
				8	8			**41710**	**1.35722**
				4	16			**33494**	**1.08987**
	2D torus	4	8	32	2	29832	281678	**156034**	**5.23042**
				16	4			**90798**	**3.04364**
				8	8			**58322**	**1.95501**
128	Ring	2	4	16	4	55308	277198	**107298**	**1.94001**
				8	8			**78574**	**1.42066**
				4	16			**64214**	**1.16103**
	2D torus	4	8	32	2	54408	502862	**278914**	**5.12634**
				16	4			**164526**	**3.02393**
				8	8			**107474**	**1.97533**

circuit for block control is needed, the hardware cost reduction effect of the memory element for control information and surrounded circuits of multiport memory exceeds the proposed method. When the router circuit is implemented on the condition of $B \leqslant C$, the cost of the proposed method becomes double to that of conventional method. As mentioned above, the hardware cost of proposed method can be reduced by "By-Block Sharing".Further hardware cost reduction is possible because arbitration and switches for the bank memory can be reduced. It is to be noted that crossbar switch is used for the shared memory of the proposed method.

6. PERFORMANCE EVALUATION - RELATION BETWEEN THE NUMBER OF BLOCKS AND COMMUNICATION PERFORMANCE

The communication performance is evaluated by software simulation. Every PE generates packet with a specified probability in every clock cycle and transmits the packet to randomly selected PE. These processes were carried out for 200000 cycles, and average transfer time and average throughput are recorded. On the same network parameters and every probability of

occurrence, simulations are carried out for 10 times, and the average of transfer time and throughput are plotted in a graph. In this experiment, the average transfer time and throughput are calculated and plotted as throughput in the horizontal axis and average transfer time in the vertical axis.

We use a dimension-order routing for packets routing to route packets. We have considered 2D-mesh and 2D-torus network of size 16 (4×4) and 64 (8×8) for performance evaluation. Two virtual channels per physical link are simulated. The message length is considered as 16, 32, and 64 flits; and the buffer length of each router is 32 and 64 bits.

We evaluate the influence of the number of blocks. If the number of blocks is small value, the hardware cost of the proposed method will become small. But, communication performance may fall because the utilization efficiency of a memory falls. Figure 6-12 portrayed the results of simulations of a torus and mesh network. The upper graphs of those figures are results of torus, and the lower are mesh.

In our evaluation, we compare the following cases;

- no-sharing：It does not share.
- by-flit-link：It is one type of link sharing method. It does not use by-block memory sharing.
- B2, B4, B8：It is a link sharing method called by-block memory sharing. In those methods, the number of blocks are 2 (B2), 4 (B4), and 8 (B8).

As shown in those figures, the following points are clarified.

1. The progress ratio of the mesh is smaller than that of torus. In a mesh network with more PEs, the ratio between edge and corner PE and total number of PE is low. Since the buffer of the unused link in corner and edge PE can be used in a mesh, the performance of mesh with a few of PE improves substantially by the proposed method. Moreover, unlike torus, since the mesh network requires one virtual channel to prevent deadlock, two channels can be freely used.

2. It is shown that the performance is significantly improved when the total amount of buffers and the packet length are similar. When packet length is very large or when the packet length is extremely small, the performance improvement is not impressive.
3. The difference in performance is trivial for B8 and by-flit-link method. On the other hand, the performance of B2 and B4 are lower than by-flit-link in many cases. As stated above, eight is enough as the number of blocks in 2D mesh and torus. Henceforth, eight is used as a basic status of the number of blocks.

4. When the total amount of buffers is larger than the packet length, a clear difference is shown between B4 and B8. Compared with packet size, the block size is too large in the

case of B4 with those situations. So the memory in a block remains. Therefore, it is thought to be desirable that the block size is smaller than packet size.

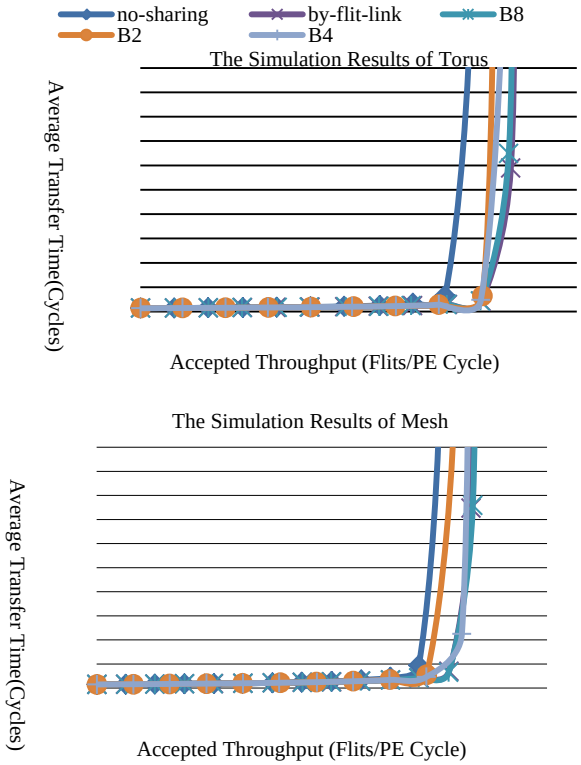

Figure 6: The communication performance of a torus and mesh: 16 PE, 32 Buffer, and 16 Flits/Packet

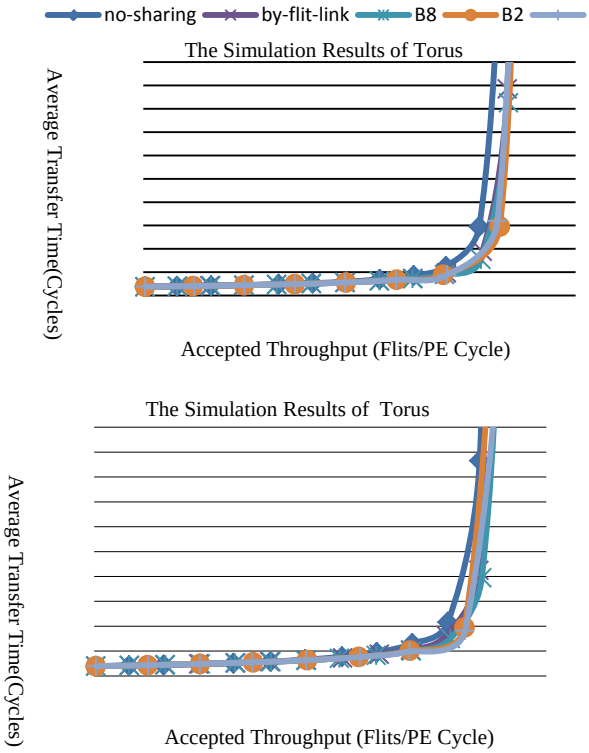

Figure 7: The communication performance of a torus and mesh: 16 PE, 32 Buffer, and 64 Flits/Packet

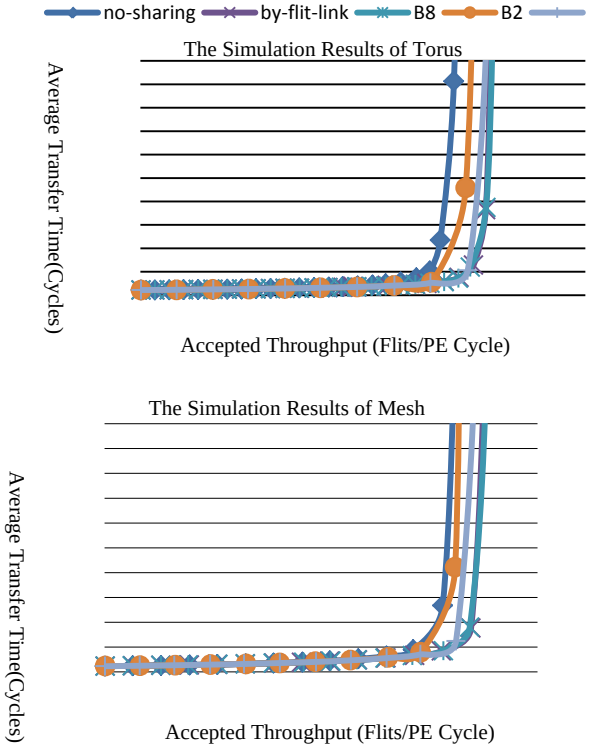

Figure 8 : The communication performance of a torus and mesh: 16 PE, 64 Buffer, and 32 Flits/Packet

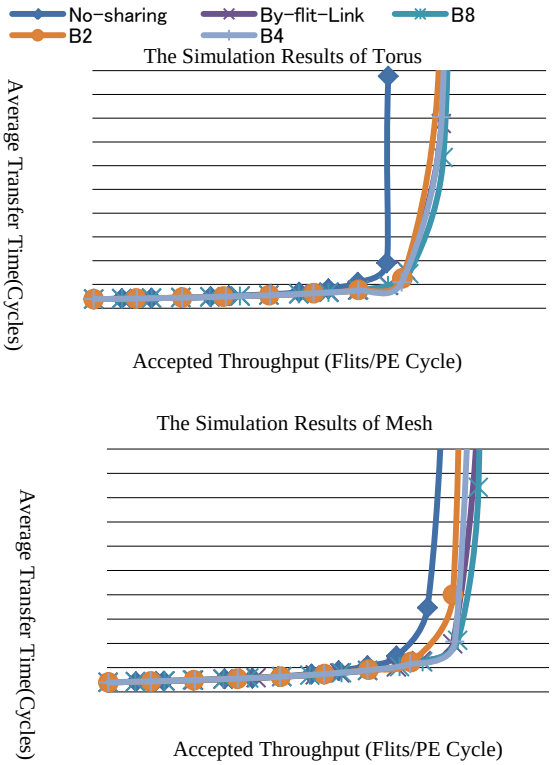

Figure 9: The communication performance of a torus and mesh: 16 PE, 64 Buffer, and 64 Flits/Packet

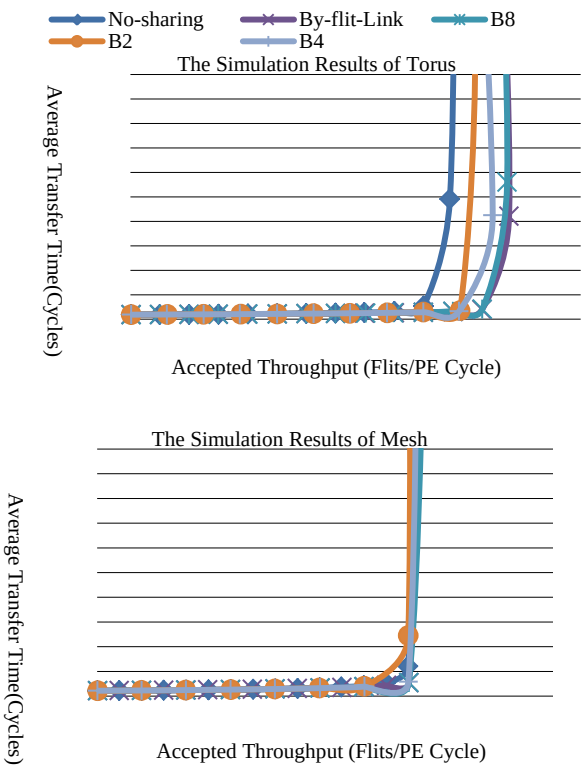

Figure 10: The communication performance of a torus and mesh: 64 PE, 32 Buffer, and 16 Flits/Packet

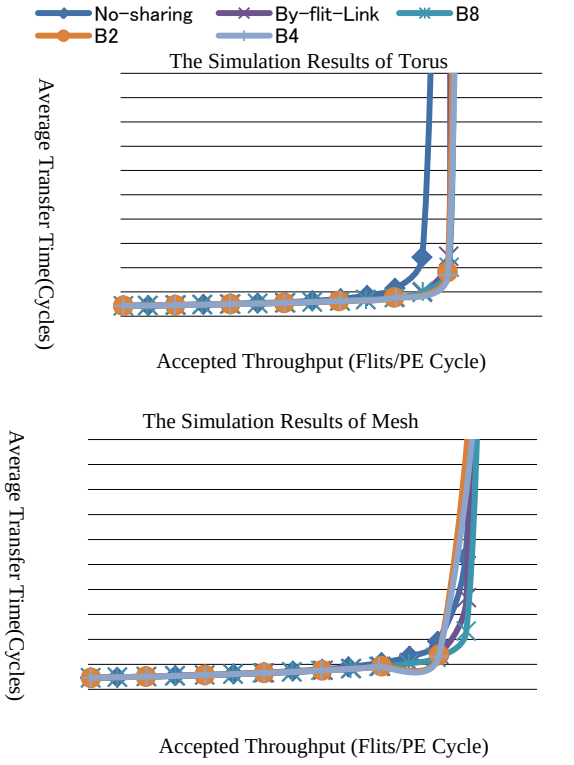

Figure 11: The communication performance of a torus and mesh: 64 PE, 32 Buffer, and 64 Flits/Packet

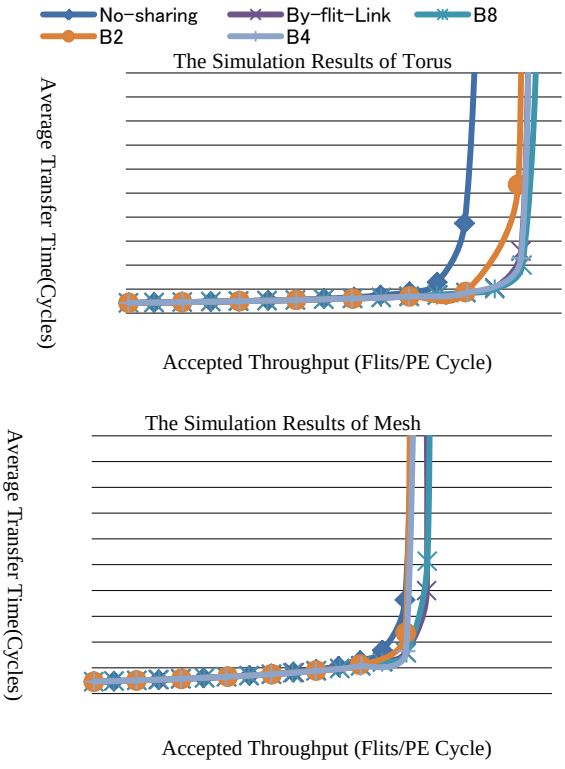

Figure 12: The communication performance of a torus and mesh: 64 PE, 64 Buffer, and 64 Flits/Packet

7. CONCLUSION

In this paper, we presented the sharing method of multiple physical links in a NoC router. And we presented the pipeline processing, method and evaluated in detail the communication performance in the different situations. The number of pipeline stages of proposed method is 2 stages larger than the traditional router in order to use a shared memory. However, we also showed that delay was concealed if the capacity of a private buffer is enough. We found that eight is sufficient enough number of banks in the multiport memory for 2D mesh and 2D-torus. We have evaluated the performance considering both 8 banks and less than 8 banks. We found that both the mesh and torus network yield the higher performance by the proposed method. Issues for future work and further exploration includes the evaluation of performance of the high dimensional network such as 3-D torus or mesh networks.

REFERENCES

[1]. Kumary, P.Kunduz, A.P.Singhx, L.-S.Pehy, N.K.Jhay, A 4.6its/s 3.6GHz single-cycle NoC router with a novel switch allocator in 65nm CMOS, 25th International Conference on Computer Design(ICCD 2007), pp.63-70, 2007.

[2]. Gregory L. Frazier, Yuval Tamir, The design and implementation of a multiqueue buffer for VLSI communication switches, Proceedings of the International Conference on Computer Design Cambridge, Massachusetts, pp.466-471, 1989.

[3]. Yuval Tamir， Gregory L. Frazier， Dynamically-Allocated Multi-Queue Buffers for VLSI Communication Switch IEEE Trans. Computers， Vol.41， No.6， pp.725-737， 1992.

[4]. Ahmadinia and A. Shahrabi, A Highly Adaptive and Efficient Router Architecture for Network-on- Chip, The Computer Journal, Vol.54 Issue 8, pp.1295-1307, 2011.

[5]. R.S. Ramanujam, V. Soteriou, B. Lin and L.S. Peh, Extending the Effective Throughput of NoCs with Distributed Shared-Buffer Routers, IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems, vol.30, No.4, pp.548-561, 2011.

[6]. Naohisa Fukase，Yasuyuki Miura，Shigeyoshi Watanabe，Link-Sharing Method of Buffer in Direct-Connection Network，The 2011 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, pp.208-213, 2011.

[7]. Naohisa Fukase, Yasuyuki Miura, Shigeyosi Watanabe, The Hardware Cost Reduction Method of Control Circuit for Link-Sharing Method of Buffer in NoC Router, 2013 RISP International Workshop on Nonlinear Circuits, Communications and Signal Processing, March 2013.

[8]. Naohisa Fukase, Yasuyuki Miura, Shigeyosi Watanabe, The Proposal of Link-Sharing Method of Buffer in NoC Router : Implementation and Communication Performance, Jounal of Basic and Applied Physics, (In printing).

[9]. Michael Golden et al., “A 500MHz write-bypassed, 88-entry, 90bit register file,” Proc. of Symposium on VLSI Technology, Session C11-1, 1999.

[10]. H.J Mattausch, K.Kishi and T.Gyohten, “Area-efficient multi-port SRAMs for on-chip data-storage withhigh random-access bandwidth and large storage capacity,” IEICE Trans. Electron., Vol.E84-C, No.3, p410, 2001.

[11]. W.J.Dally, Virtual-Channel Flow Control, IEEE Trans on Parallel and Destributed Systems, Vol. 3, No. 2, 1992.

[12]. M. Ni and P. K. McKinley， A Survey of Wormhole Routing Techniques in Direct Networks， Proc of the IEEE， Vol. 81， No. 2， pp. 62-76， 1993.

[13]. E. Fleury and P.Fraigniaud, A General Theory for Deadlock Avoidance in Wormhole-Routing Networks, IEEE Trans. Parallel and Distributed Systems, Vol. 9, No. 7, pp. 626-638, 1998.