

Link-Sharing Method of Buffer in NoC Router: Its Implementation and Communication Performance

Naohisa Fukase^{*1}, Yasuyuki Miura², Shigeyoshi Watanabe³

Shonan Institute of Technology, Fujisawa, Kanagawa, Japan

^{*1}12t2501@sit.shonan-it.ac.jp; ²miu@info.shonan-it.ac.jp; ³watanabe@info.shonan-it.ac.jp

Abstract- The sharing method is proposed that a memory is shared between plural physical link by using the multi-port memory in wormhole router for NoC (Network on Chip) as efficient usage of the flit buffers. If the proposed method is implemented by conventional method, hardware costs will increase greatly. Therefore, sharing a memory by block size is put forward. This paper presents the outline and pipeline processing of the proposed method. The pipeline of the proposed method has two courses of the routes 1 and 2. The number of pipeline stages of route 2 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. In addition, the effectivity of proposed method is shown by simple estimation of hardware cost and evaluation result of communication performance by software simulation.

Keywords- Router; Interconnection-Network; Network on Chip; Multi-Port Memory

I. INTRODUCTION

Recently, many studies have been carried out on NoC (Network on Chip) which connects cores (IP core, circuit block, PE, memory, processor) by a network on a chip. Many advantages of NoC have been discovered, but there were some problems such as the increase of layout area and the power consumption. Therefore it is necessary to design a high-performance router in a limited hardware resource in NoC.

To utilize buffers in the router efficiently, some methods to share one memory in plural virtual channels [1] have been proposed and implemented [2, 3]. However, it was only the sharing of a few virtual channels. Therefore, the buffer sharing method of plural physical links is proposed, by which much more channels can be shared and the router circuit can utilize buffers more efficiently.

Similar to the proposed method in this paper, the method of sharing a ring buffer [4], and the method that some buffers are shared between some channels [5] are also proposed. Since the former method use the crossbar switch of a large size, ring buffer of a large size is difficult; the latter method does not use wormhole routing [6], and since the number of pipeline stages increases, the communication latency becomes larger.

Now, the method of sharing a buffer by plural physical links in a router is proposed to use a buffer effectively [7]. If the method is implemented by the traditional method, the hardware cost will increase with the implementation of sharing a large number of physical links. In order to solve the problem, a Multi-bank Multiport Memory is adopted to reduce the hardware cost [8-10].

In this paper, after the explanation of the conventional method in section II the outline of proposed method is introduced in section III, then the pipeline processing flow is presented in section IV, and software simulation is carried out to simply estimate the hardware cost and communication performance in order to show the efficiency of the proposed method in section V and VI; finally an conclusion is made in section VII.

II. CONVENTIONAL METHOD

The structure of the PE (Processing Element) is shown in Fig. 1. As shown in Fig. 1, a PE consists of one or more processor cores and a router circuit. In the router circuit, a crossbar switch connects links where communication course is laid out. A link usually has plural virtual channels to smooth the flow of packets, and a buffer is arranged to each channel of the input side of the crossbar switch for smoothing the communication. To reduce the hardware cost of a PC, wormhole routing [1] which can be implemented by using a little buffer is used in NoC.

In wormhole routers of simple structure, a buffer of the same capacity is installed to each channel [1, 11]. By such a structure, there is a problem that "a buffer allocated to the channel is not utilized effectively". To solve this problem, the technique to share a memory by flits between plural virtual channels in a physical link is proposed and implemented conventionally [2, 3].

The structure of the router in this method is shown in Fig. 2. By this method, memory region from a shared memory is assigned dynamically and used when the capacity of the buffer with each channel becomes insufficient. In this method, a connection between acquired memory regions is expressed by recording the assigned memory region number to "VC Block info" of the assigning channel. In this paper such method is called "Channel Sharing" and the proposed method mentioned in

the next section is called “Link Sharing”.

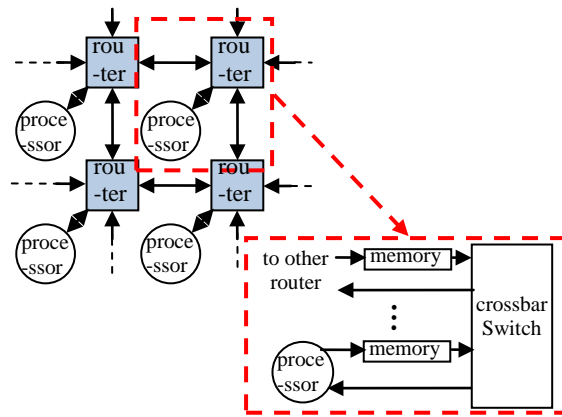


Fig. 1 PE structure on NoC

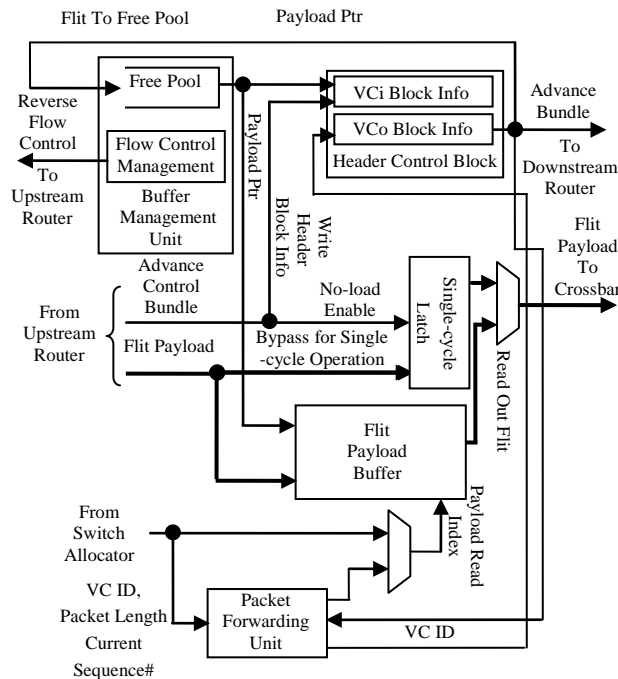


Fig. 2 Router structure of conventional method

III. PROPOSED METHOD

The increase of hardware costs is attributed to that the sharing method of the buffer over a physical link still hasn't been put into practice until now. Because it responds to the concurrent access from some physical links, the link sharing method needs to use a multi-port memory for memory sharing. However, the hardware cost increases by use of the normal multiport memory.

The structure of the proposed method and the multi-bank multiport memory used in our method are depicted in Fig. 3 and Fig. 4, respectively. As portrayed in Fig. 3, 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 Fig. 4, 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”.

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 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 [12, 13].

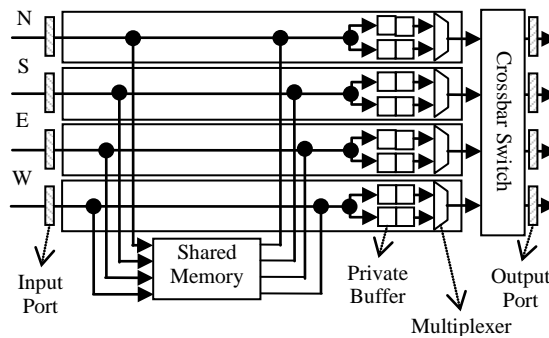


Fig. 3 Router structure of proposed method

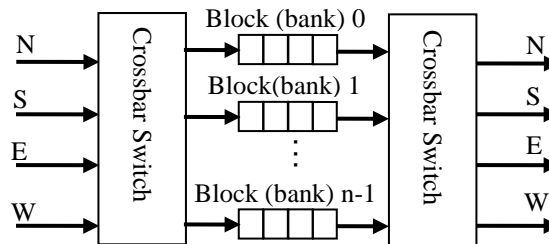


Fig. 4 Multi-bank multiport memory

IV. PIPELINE STRUCTURE

A. Structure of Traditional Router Pipeline

The pipeline structure of the traditional router is shown in Fig. 5 [14]. As shown in Fig. 5, 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.

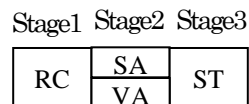


Fig. 5 Structure of the traditional router pipeline

B. Pipeline Structure of Proposed Method

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

route 1: input port \Rightarrow private buffer \Rightarrow output port

route 2: input port \Rightarrow shared memory \Rightarrow private buffer \Rightarrow output port

Route 1 is a course immediately sent to Private Buffer after arriving at an input port. The pipeline of route 1 is shown in Fig. 6. As shown in Fig. 6, 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 Fig. 6. 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.

Route 2 is a course that the packet is sent into the shared memory and then goes to the private buffer after that. 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 route 2 is shown in Fig. 7. Route 2 needs the following stages:

1) IJ (In-Judge)

“Whether the shared memory is used”, and “A now 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 Fig. 7, the number of stages of route 2 is 2 stages larger than route 1. But the delay by the pipeline of route 2 is concealed by following reasons.

- When a network is not crowded and the private buffer is not full, it becomes the 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 sent to a crossbar switch from the private buffer. 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 Fig. 8. In Fig. 8, the 3rd flit goes to route 2 since top flit was blocked.

Stage1	Stage2	Stage3
RC	SA	ST
IJ	VA	

Fig. 6 Route 1 pipeline

Stage1	Stage2	Stage3	Stage4	Stage5
IJ	SiA	SiT	SA	ST
		SoA	SoT	

Fig. 7 Route 2 pipeline

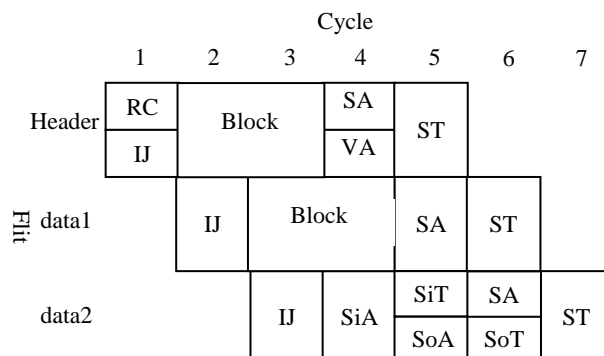


Fig. 8 The example of the proposed pipeline

C. Hardware Structure

A block diagram including the pipeline structure of the proposed method is portrayed in Fig. 9. As depicted in Fig. 9, the proposed method has 5 pipeline stages. Each stage is the area surrounded by the dashed line of Fig. 9. Each stage is divided by the pipeline register (shown by rectangle in the figure) and buffers such as shared memory and private buffer. The packet which passes route 2 uses all the stages in Fig. 9, and the packet which passes route 1 uses three stages which pass through the 2nd and 3rd stage.

The structure of the control circuit of proposed method is shown in Fig. 10. In this method, the memory is controlled by two kinds of queue “Free Pool” and “Block info”. Free Pool is the queue to store pointers which are not reserved by any channels. Block info is provided for each channel. This is the queue to store the pointers of memory which are reserved by each channel. When a channel reserve the memory, the pointer stored at head of Free Pool is input to the tail of Block info of the channel. When a channel frees the memory, the pointer stored at head of Block info of the channel is input to the tail of Free Pool.

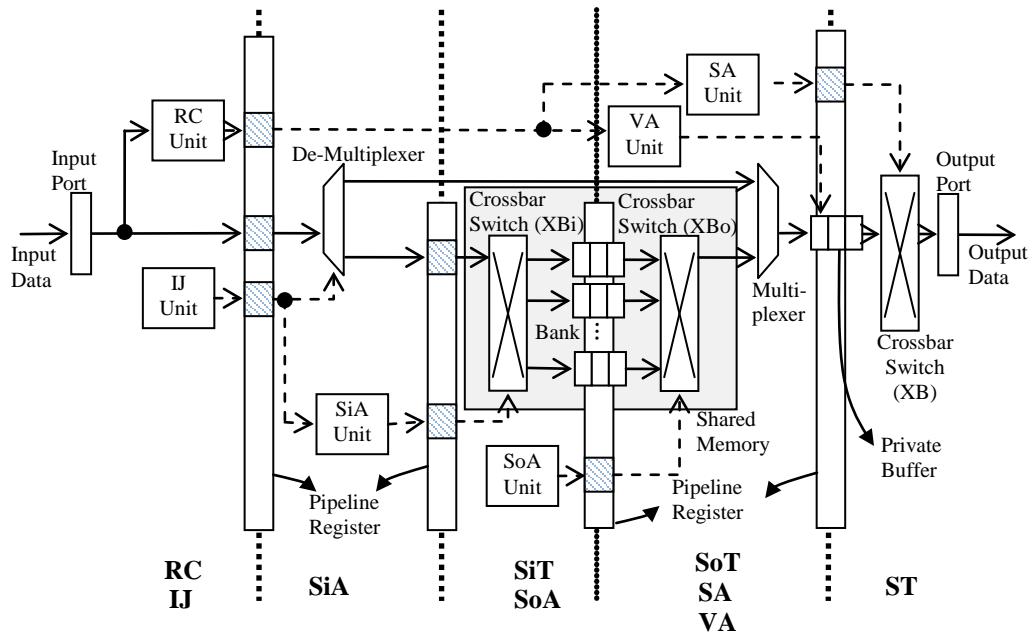


Fig. 9 The block diagram of proposed method

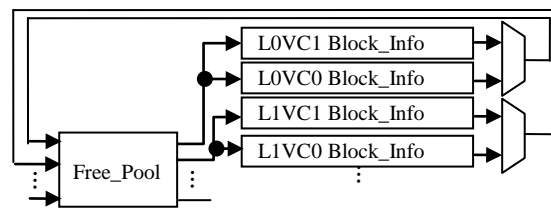


Fig. 10 The controller of proposed method

V. 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 “a) buffer” of the physical link except crossbar switch and control circuit.

“b) Memory element for control information” is needed for both the traditional method (the method that a memory is shared by plural virtual channels) and the proposed method. b) in above includes the Free Pool and Block_Info for control the shared memory.

Additional hardware costs for the proposed method are “c) logic circuit for block control” and “d) 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 $2n$, 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 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 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 $B \leq C$, 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.

VI. COMMUNICATION PERFORMANCE

The communication performance is evaluated by software simulation. To evaluate the performance, we analysed by a sharing range (the channel sharing or the link sharing) and the sharing candidate (by-flit or by-block). For evaluation, we assumed 64-PE bi-directional 2D-torus which has 4 links, 2 channels in each link. The total size of buffer in a router is 512, and the number of flits in a packet is 512. The routing algorithm uses dimension order routing.

In this experiment, vertical axis of the graph is the average transfer time for packet and the horizontal axis are the average throughput. The random packet whose destination PE is randomly selected and whose request probability is P were transmitted from each PE. Then we recorded the throughput and the packet transfer time, and we plotted average values of them to the graph. Several times of the above process with changing P were carried out, and graphs were made.

The result of conventional “channel sharing” which shares a memory by plural channels in a physical link and the result of proposed method “link sharing” which shares a memory by all physical links are shown in Fig. 11. As shown in Fig. 11, it was able to confirm that performance improved by “link sharing” than “channel sharing”.

TABLE 1 IMPLEMENTATION COST OF PROPOSED METHOD (TRANSISTORS)

W	Topo logy	L	C	B	F	Conventional Method	By Flit and Link Sharing	By block and Link Sharing(Proposed Method)					
								Buffer	Control Memory	Control Logic	Memory Surround	Total	Conventional /Proposed
64	Ring	2	4	16	4	30732	154318	24576	2670	5428	25472	58146	1.89203
				8	8			24576	1266	3228	12640	41710	1.35722
				4	16			24576	606	2040	6272	33494	1.08987
	2D torus	4	8	32	2	29832	281678	24576	9810	18992	102656	156034	5.23042
				16	4			24576	4422	10856	50944	90798	3.04364
				8	8			24576	2010	6456	25280	58322	1.95501
	3D torus	6	12	48	2	44748	639786	36864	24342	48240	233856	343302	7.6719
				24	4			36864	10938	26724	115968	190494	4.25704
				12	8			36864	4950	15276	57504	114594	2.56087
128	Ring	2	4	16	4	55308	277198	49152	2670	5428	50048	107298	1.94001
				8	8			49152	1266	3228	24928	78574	1.42066
				4	16			49152	606	2040	12416	64214	1.16103
	2D torus	4	8	32	2	54408	502862	49152	9810	18992	200960	278914	5.12634
				16	4			49152	4422	10856	100096	164526	3.02393
				8	8			49152	2010	6456	49856	107474	1.97533
	3D torus	6	12	48	2	81612	1119018	73728	24342	48240	455040	601350	7.3684
				24	4			73728	10938	26724	226560	337950	4.14094
				12	8			73728	4950	15276	112800	206754	2.53338

A result when a memory is shared by using “by-flit” sharing and a result by using “by-block” sharing are shown in Fig. 12. As shown in Fig. 12, the noticeable difference was not seen in performance of both methods. Although “uniform” communication that the destination PE is randomly selected is carried out in this experiment, actually the channel selection method is different with wraparound channels and the other channels. In addition, sometimes the frequency of channel used is different from direction channels and direction channels. Therefore the frequency of channel used is different in the dimension order routing. As a result, specific channels tend to acquire memory. So it is thought that a noticeable difference does not appear in the communication performance between “by-block sharing” that memory area are collectively acquired by the channel and “by-flit sharing” that memory area are acquired flit by flit.

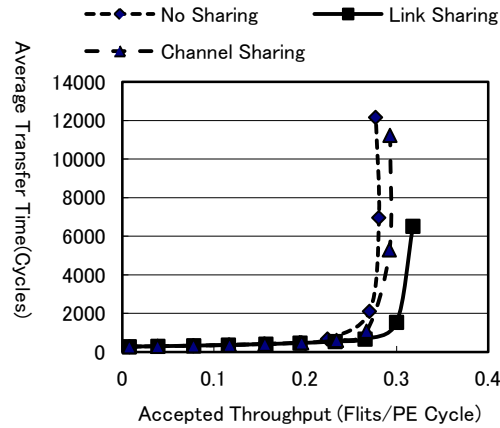


Fig. 11 The communication performance (conventional channel sharing and link sharing)

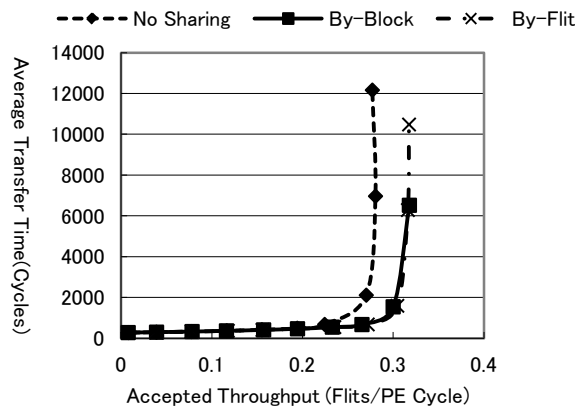


Fig. 12 The communication performance (by-flit sharing and by-block sharing)

VII. CONCLUSIONS

In this paper, we proposed the buffer sharing method of plural physical links in NoC router. If proposed method is implemented by the traditional implementation method, increase of the hardware cost with the implementation becomes a problem. So we proposed a method of “by-block sharing”. Furthermore, the pipeline structure of the proposed method is presented. The pipeline of this method has two courses of routes 1 and 2. The number of pipeline stages of route 2 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. The hardware cost and communication performance were evaluated. As a result, it was shown that the implementation with a little hardware-cost and high communication performance can be achieved by “by-block” implementation.

REFERENCES

- [1] W. J. Dally, “Virtual-channel flow control,” IEEE Transaction on Parallel and Distributed Systems, vol. 3, No. 2, 1992.
- [2] Kumary, P. Kunduz, A. P. Singhx, L. -S. Pehy and N. K. Jhay, “A 4.6Tbits/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.
- [3] Gregory L. Frazier and 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.
- [4] A. Ahmadiania and A. Shahrabi, “A highly adaptive and efficient router architecture for network-on- chip,” The Computer Journal, vol. 54(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(4), pp. 548-561, 2011.
- [6] M. Ni and P. K. McKinley, "A Survey of Wormhole Routing Techniques in Direct Networks," Proc of the IEEE, vol. 81(2), pp. 62-76, 1993.
- [7] Naohisa Fukase, Yasuyuki Miura and 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.
- [8] Y. Tatsumi et al., "Fast quadratic increase of multiport-strage-cell area with port number," Electronics Letters, vol. 35(25), pp. 2185-2187, 1999.
- [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 with high random-access bandwidth and large storage capacity," IEICE Trans. Electron., vol. E84-C(3), p. 410, 2001.
- [11] W. J. Dally and H. Aoki, "Deadlock-free adaptive routing in multicomputer networks using virtual channels", IEEE Trans. on Parallel and Distributed Systems, vol. 4(4), pp. 466-475, 1993.
- [12] E. Fleury and P. Fraigniaud, "A General Theory for Deadlock Avoidance in Wormhole-Routing Networks," IEEE Trans. Parallel and Distributed Systems, vol. 9(7), pp. 626-638, 1998.
- [13] M. Koibuchi, K. Anjo, Y. Yamada, A. Jouraku and H. Amano, "A Simple Data Transfer Technique Using Local Address for Networks-on-Chips," IEEE Transaction on Parallel and Distributed Systems, vol. 17(12), pp. 1425-1437, 2006.
- [14] W. J. Dally and B. Towles. Principles and practices of interconnection networks, Morgan Kaufmann Publishers, 2004.



Naohisa Fukase was born in 1986. He received the bachelor's degree in Shonan Institute of Technology (SIT) in 2010, and the MS degree in SIT in 2012. He is a PhD course student of graduate school of SIT. His research interest is the interconnection network of Network on Chip.



Yasuyuki Miura received the bachelor's degree in Tohoku University in 1997, and the MS and PhD degree in Japan Advanced Institute of Science and Technology (JAIST) in 1999 and 2002. Then he had worked in the National Institute of Communication Technology (NICT), Japan until December 2004. From January 2005 to March 2006, he was researcher of the Japan Science and Technology Agency (JST). Since April 2006, he was lecturer in the Shonan Institute of Technology. Since April 2011, he is associate professor. His research interests include parallel processing, interconnection network, and MPEG video streaming. He is a member of the IEEE and the IEEE Computer Society, and IPSJ of Japan.



Shigeyoshi Watanabe received his BSc. degree in Keio University in 1977, and the MS degree in Tokyo Institute of Technology in 1979. Then he had worked in the Toshiba Corporation Limited. His research includes circuit design of Non-volatile memory and DRAM, CMOS/ Bi-CMOS/SOI/3-dementional transistor devices, and high-speed, low-power circuit and archi-tecture for system LSI. Since 2005, he is professor in the Shonan Institute of Technology. He has PhD degree of technology.