Original scientific paper

Informacije MIDEM
Journal of Microelectronics,
Electronic Components and Materials
Vol. 43, No. 2(2013), 111 – 118

A dynamic adaptive arbiter for Network-on-Chip

Yanhua Liu[1,2], Jie Jin[2,3], Zongsheng Lai[1]

[1]Institute of Microelectronics Circuit & System, East China Normal University, Shanghai, China
[2]Jiangsu Provincial Key Lab of ASIC Design, Nantong University, Nantong, China
[3]Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, China

Abstract: Network-on-chip (NoC) is considered as a promising paradigm to overcome the communication bottleneck of future multicore systems. As a basic component in on-chip router, arbiter has a big impact on the performance of router. In this paper, we propose a novel dynamically adaptive arbiter which is based on the round robin mechanism. The proposed arbiter detects buffer status of input ports and changes priorities of the input port dynamically to enhance the performance of the router. Simulation results show that the proposed arbiter can achieve 7.3% improvement in saturation packet injection rate and 13.3% improvement in saturation throughput of NoC on average, when compared with round robin arbiter. Using Synopsys design tools with 0.18-μm technology, implementation results show that a router with the proposed arbiter needs additional 4.8% area compared to a router with round robin arbiter.

Key words: Network-on-chip (NoC), on-chip router, dynamically adaptive arbiter, round robin mechanism,

Dinamično adaptiven razsodnik za omrežje na čipu

Povzetek: Omrežje na čipu (NoC) se smatra za obetajoč zgled pri premagovanju ovir v bodočih večjedrnih sistemih. Razsodnik, kot glavni element na usmerjevalniku na čipu, ima velik vpliv na učinek usmerjevalnika. V članku je predlagan nov dinamično prilagodljiv razsodnik, ki temelji na mehanizmu krožnega dodeljevanja. Predlagan razsodnik zazna status medpomnilnikov vhodov in dinamično spreminja prioritete vhodov tako da poveča učinek usmerjevalnika. Simulacije prikazujejo, da predlagan razsodnik omogoča povprečno 7.3 % povečanje učinkovitosti pri stopnji injekcije nasičenih paketov in 13.3 % povečanje pri prehodu NoC v primerjavi z delovanjem v krožnem dodeljevanju. Pri uporabi načrtovalskega orodja Synopsys v 0.18 μm tehnologiji predlagani razsodnik potrebuje 4.8 % več prostora.

Ključne besede: omrežje na čipu, usmerjevalnik na čipu, dinamično adaptiven razsodnik, mehanizem krožnega dodeljevanja

** Corresponding Author's e-mail: liu8033@163.com*

1 Introduction

Traditional bus-based communication architecture has been proved to be a bottleneck that limits the scalability, reusability and reliability of future System-on-Chip (SoC). Networks-on-chip (NoC) has been proposed as a new communication concept that satisfies requirement of the future SoC [1]. NoC provides technologies for generic on-chip interconnection network realized by routers which connects processing elements (PEs) such as ASICs, FPGAs, memories and IP cores [2, 3]. The communication data of PEs are packetized and transferred through the on-chip network. NoC has significant advantages on performance, reusability and scalability compared to traditional bus-based architecture.

On-chip router is the most dominant component of the high-performance NoC. There are three major aspects in design and implementation of a high-performance router: 1) an improving routing algorithm that provides conflict-free paths between input and output ports, 2) an efficient arbiter that authorizes the requiring input ports based on good schedule mechanism, and 3) an optimal buffer distribution that holds more packets in input ports. Although the routing paths are determined by route computation in the router, connection between input and output ports is implemented by arbiter. Arbiter can determine implementation sequence of the routing paths if there exist some conflict requests for the same output port. Buffer can house the incoming packets that cannot be immediately forwarded due

to output port contention or blocking. Increasing buffer slots for heavy-load input ports can improve performance of router. However, more buffer slots will cause more area and power consumption. If heavy-load input ports can be authorized with high priority when they have requests for connecting output port, the pressure on these input port will be decreased and the performance of the router can be improved. So, among the three major aspects, arbiter plays an important role to enhance the router performance.

In this paper, we propose a novel dynamically adaptive arbiter (DAA) which can change the priority of input port dynamically according to its buffer status. Buffer full signal is used as high priority signal of input port. The input port with high priority is allowed to occupy its desired output port prior to other ports, and thus the buffer pressure of the input port can be decreased efficiently. Simulation results show that NoC using proposed arbiter achieves higher performance.

This paper is organized as follows: Section 2 describes related works for arbiters of on-chip router. Section 3 describes details of the proposed arbiter. Section 4 gives experimental and comparison results. Finally, conclusions are made in section 5.

2 Related Works

Many researchers focused on developing various arbitration schemes in order to achieve an efficient allocation and reduce packet latency. A lot of arbiters have been proposed in the routers of computer network such as round robin arbiter [4], fixed priority arbiter [5], lottery arbiter [6], token ring arbiter [7] and so on.

Round robin arbiter treats each input port fairly and guarantees fairness in scheduling. Using round robin arbiter, each input port have an equal chance to own the output port and the starvation problem can be solved. However, round robin arbiter is too fair and may cause low efficiency for some input ports. Fixed priority arbiter always authorizes the requiring input port with the highest priority when requiring contention happens. The input ports with lower priority may rarely be authorized which results in extremely unfair. Lottery arbiter offers input ports certain numbers of lottery as their priority level. Input port with more lotteries has bigger probability to win the output port. However, if the number of lotteries is static, some input ports which have little lotteries may be hardly responded under heavy traffic load. Token ring arbiter cannot guarantee correctness and may miss some requests from different input ports.

All above arbiters can be used in computer network but they are unfit for on-chip router. For on-chip router, resource consumption, average packet latency and complexity of priority strategy should be considered. Recently, some new arbitration methods based on round robin and lottery mechanisms were proposed for the on-chip router. A priority based output arbiter was proposed in [8] which could eliminate the congestion state of NoC. Before arbitrating, the arbiter counted the number of output port requirements for packets in each input port and gave a higher priority to the input port which had more requirements. Zhu et al. [9] presented three new scheduling methods based on round robin mechanism for the on-chip router. The three scheduling methods used different heuristic information to determine the scheduling sequence. However, these arbitration mechanisms needed to detect all packets in input ports and compute their routing paths before delivering them. This resulted in difficult hardware implementation of control logics and totally unsuitable for on-chip router.

Zhang [10] designed a statistic-based lottery arbiter which did not cause starvation problem. However, this arbiter needed many registers, which leaded to a large amount of resource consumption. A customized priority arbiter based on lottery mechanism for the on-chip router was proposed by Wu et al. [11]. The arbitral priorities were customized according to the communication cases among PEs in NoC. This arbiter used static priority which was not fit for dynamic real applications in NoC. Wang [12] improved the lottery arbiter and presented a dynamic lottery arbiter. The dynamic lottery arbiter detected the loads of input ports in every clock cycle and adjusted the priority of each input port dynamically. The proposed arbiter did not work efficiently under the uniform traffic. Even under non-uniform traffics, some heavy-load input ports might always occupy the output ports which caused starvation problem.

3 Design of dynamic adaptive arbiter

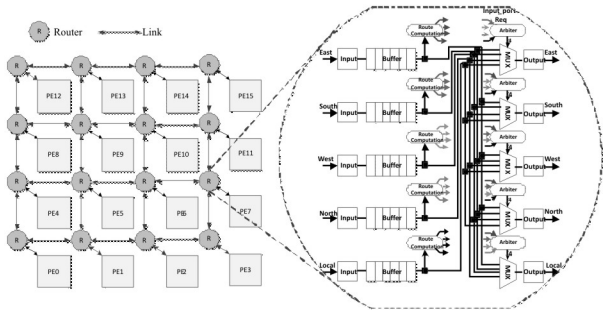
In our proposed arbiter, the input buffer full signals are detected as the high priority signals. A high priority will be given to the input port if the buffer of port is full. In order to prevent packet starvation, we design the proposed arbiter based on round robin mechanism. In this section, we first present a typical NoC platform and head-of-line blocking problem. Then, we describe the working principle of the proposed arbiter.

3.1 NoC platform and head-of-line blocking problem

Fig. l shows a typical 2D mesh NoC and the corresponding router architecture. 2D mesh is the most popular

topology for Network-on-Chip which has good scalability. The communication data named packet can be transferred by the on-chip routers and links in 2D mesh NoC. On-chip router is the core component in the NoC and has big impact on system performance [13].

Wormhole router is one of the most commonly used on-chip routers in NoC. It is easy for implementing and suitable for on-chip network. A typical wormhole on-chip router with dedicated buffer per input port is shown in fig. 1 (b). In the wormhole router, each packet is divided into small unit called flit. Head flit proceeds through all the stages while body and tail flits skip route computation and output arbitration stages. Body and tail flits inherit the output port allocated to the head flit. The last flit in a packet, called tail flit, releases the reserved output port that have been reserved by the header flit of that packet, when it departs the current router. The route computation determines deliver direction of the packet according to the destination of the head flit and the routing algorithm. Route computation sends request to the arbiter after determining deliver direction. Arbiter grants the request and connects output port with input port by MUX.



(a) (b)

Figure 1: (a) 2D mesh NoC topology, (b) wormhole router architecture

Arbiter is one of the key components in the on-chip router. It can determine the output sequence when output contention happens. If packets arrive at different input ports but need to be dispatched into the same output port simultaneously, a output contention will happen. In order to solve the contention, an arbitration mechanism is necessary to allow only one input port to access the output port. Most arbiters are unconcerned with buffer status and authorize the input port by a determinate mechanism. These arbiters may cause head-of-line blocking problems if the input port with full buffer cannot be authorized preferentially. Fig.2 shows an example of head-of-line blocking problem. As shown in fig. 2, west and south input ports request for east output port simultaneously in router R2. If the east output port is connected to the south instead of west input port in R2, packet p1 in west input buffer of R1 cannot advance. The packets behind p1 will occupy west input buffer in R1 and decrease the network performance. Therefore, head-of-line blocking is a key factor when evaluating different arbiters for on-chip router.

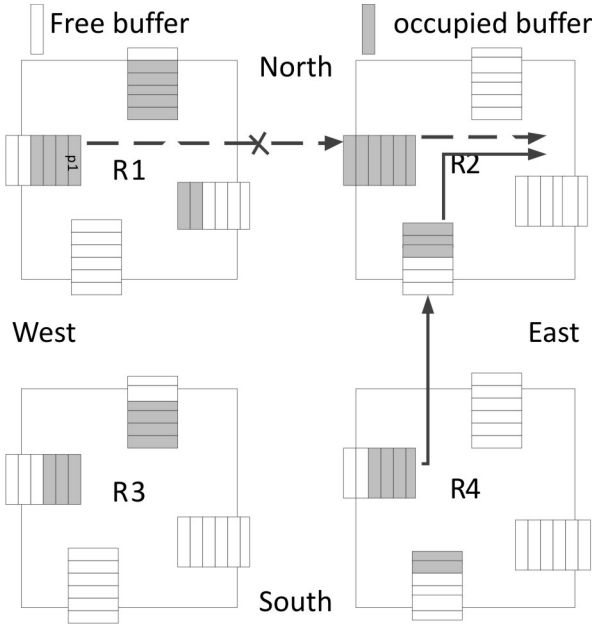


Figure 2: Head-of-line blocking problem induced by inefficient arbitration

3.2 Architecture of dynamically adaptive arbiter

In order to alleviate head-of-line blocking problem, we propose a dynamic arbitration priority for each input port according to the buffer status of the port. If some input ports request the same output port simultaneously, the proposed arbiter will detect the buffer status of these input ports and check whether their buffers are full. Heavy-load input ports makes their buffer easily full. An input port with a full buffer means it cannot hold incoming packets any more. If packets in the full buffer cannot be delivered as soon as possible, the packets will be halted and cause packet latency. For this reason, the input port with full buffer should be authorized with high priority. However, under heavy-load traffic distribution in NoC, there is always more than one input port in a router whose buffer is full. If these input ports request the same output port, these ports should be treated fairly and have an equal chance to win the desired output port. Under the light-load distribution in NoC, there may be no input port in a router whose buffer is full. In other words, no input port has high priority and can be authorized preferentially. So, all input ports should be authorized with an equal chance. Fig. 3 shows a block diagram of one output arbiter in on-chip router which uses the proposed arbitration mecha-

nism. In fig.3, Buffer_full and Input_Port_Req are buffer full signals and request signals from input port of different directions.

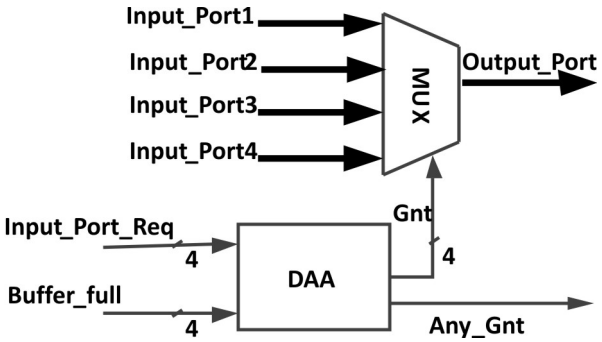


Figure 3: Block diagram for one output port arbiter

Fig. 4(a) gives detailed structure of DAA. As shown in fig. 4(a), DAA is based on two round robin arbiters. In the previous NoC research, many on-chip routers used round robin arbiter as their output arbiter [2] [14]. Round robin arbiter performs well for uniform distribution traffic, but it is not flexible for unbalanced distribution traffics when there are hotspots or priority requirements in NoC. In DAA, if some requests for output ports are generated by the input ports with full buffers, high priorities will be granted to these requests. These input ports will be scheduled first by round robin arbiter1. Other request of input ports with low priorities will be disabled when the high priority ports are scheduled.

In order to prevent starvation for the low priority input ports, a counter and a comparator are used in DAA. The counter is used to record the number of authorizing times for high priority input ports. If the number of authorizing times is bigger than T_threshold, all input ports should be scheduled with equal priority by round robin arbiter2. T_threshold can be customized by NoC designer according to the characteristics of traffics. Moreover, it can be found that, when one of the round robin arbiters works, the request signals for the other round robin arbiter will be disabled. This will ensure that no request signal is missed by DAA in each clock cycle.

Fig. 4(b) shows the typical block diagram of the round robin arbiters. It mainly consists of two barrel shifters, one simple arbiter and one shifter pointer coder. Fig. 4(c) and (d) give the hardware structure of the barrel shifter1 and simple arbiter, respectively. The barrel shifter2 has a similar hardware structure with barrel shifter1 but offers opposite shift direction. The shifter pointer coder generates the shift pointer for two barrel shifters according the previous grant. The detailed working mechanism of the round robin arbiter will not described in this paper because it has been researched

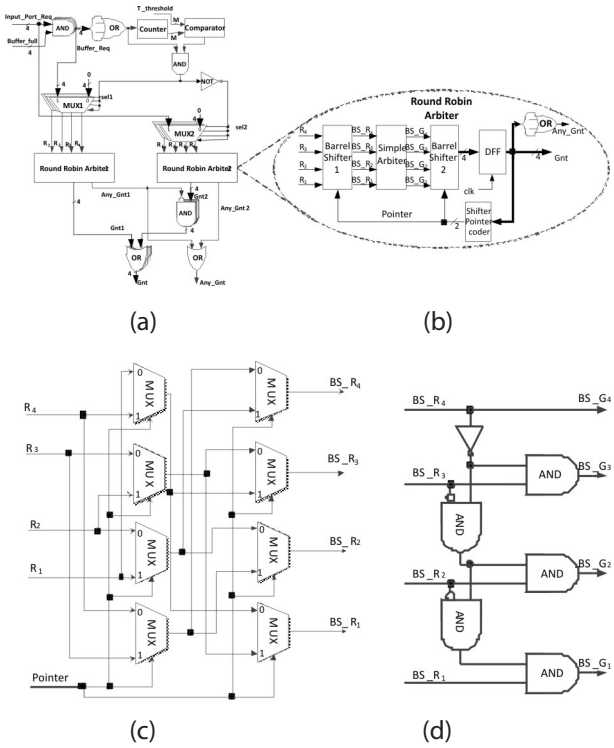


Figure 4: (a) Detailed structure of DAA (b) block diagram of round robin arbiter (c) hardware structure of barrel shifter1 (d) hardware structure of simple arbiter

in many papers [15] [16] [17]. For clarity, the description of input ports scheduling by DAA is illustrated in fig. 5 by pseudo-C language.

Following the scheduling method described in fig. 5, DAA assigns high priorities for input ports whose buffers are full, and priorities are changed dynamically according to the Buffer_full signal. When a buffer of input port is not full, the corresponding buffer full signal will be disabled, and then the input port will lose its high priority. However, with the advance of time, some light-load input ports will accumulate more and more packets if they have no chance to occupy the desired output port. When buffers of these input ports become full, they will be given high priority to connect with their desired output ports. Even for some input ports whose buffers are never full, they will also have a chance to occupy their desired output port by the appropriate T_threshold. This scheduling method avoids the starvation problem thoroughly.

4 Results

4.1 Performance evaluation

In this section, A cycle-accurate NoC simulator implemented in SystemC is used to evaluate the performance

```
input: Input_Port_Req[4], Buffer_full[4], T_threshold[M];
output: Gnt[4], Any_Gnt;
static int count;
Buffer_Req=Input_Port_Req&Buffer_full;
if (Buffer_Req! = 0 & & count < T_threshold)
  { count = count + 1; Gnt = 0;
    for (j = 1; j <= 4; j + +)  / / scheduled by round robin arbiter1
    {k1=(last_grant_1+j) mod 4; // start from the request which is
     //the first next to last granted one
       if (Buffer_Req[k1]==1) // Rk1 have a request
        { Gnt[k1]=1; Any_Gnt=1; last_grant_1=k1; break;}
     }
  }
else if ((Buffer_Req==0||count==T_threshold )&&Input_Port_Req!=0 )
  { if (count==T_threshold) count=0;    Gnt=0;
    for (j = 1;j <= 4; j++) // scheduled by round robin arbiter2
     { k2=(last_grant_2+j) mod 4;
       if (Input_Port_Req[k2]==1)
       { Gnt[k2]=1; Any_Gnt=1; last_grant_2=k2; break;}
      }
   }
else {Gnt = 0;  Any_Gnt = 0;} / /no input port is authorized
```

Figure 5: the scheduling method of DAA

of DAA for different traffics. In order to demonstrate the effectiveness of DAA, latency and throughput are chosen as performance metrics of NoC. We compare the performance of NoC based on round robin arbiter [4] (RRA-NoC), lottery arbiter (LA-NoC) [12] and the proposed arbiter (DAA-NoC). All simulations are carried in a 4×4 mesh network for 20000 cycles with X-Y routing and wormhole switching. Four traffic patterns are simulated including three synthetic traffic patterns (Uniform, Bit-complement and Transpose) [18] and one real benchmark (VOPD application) [19]. For synthetic traffic patterns, each packet contains 4~8 flits randomly and for real benchmark, the number of flits in each packet is determined by the bandwidth requirement of the VOPD application.

Latency is defined as the time (in clock cycles) that elapses between the occurrence of a header flit injection into a network at the source node and the occurrence of a tail flit reception at the destination node. We use the average latency, L, as a performance metric like follows:

$$L = \frac{1}{K}\sum_{i=1}^{K} L_i \tag{1}$$

where K is the total number of packets reaching their destination nodes and L_i is the latency (cycles) of packet i. We define throughput TP as follows:

$$TP = \frac{total\ received\ flits}{number\ of\ PEs \cdot total\ cycles} \tag{2}$$

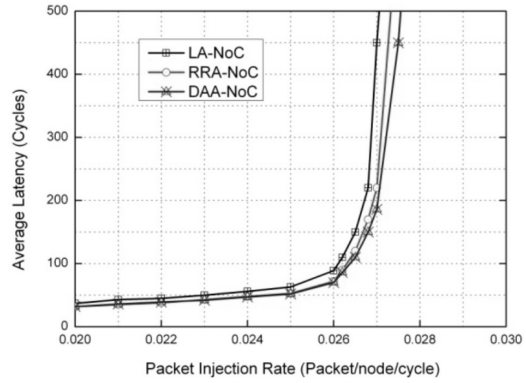
where total received flits is the number of flits that arrive at their destination nodes during the simulation, and total cycles is the number of clock cycles elapsed between the start and the end of simulation. Throughput can be used to measure the maximum traffic load that the NoC can handle.

For each traffic scenario, we first give the average packet latency with various packet injection rates (PIRs). Fig. 6 shows the packet latencies for NoCs with different arbiters under the four traffic patterns.

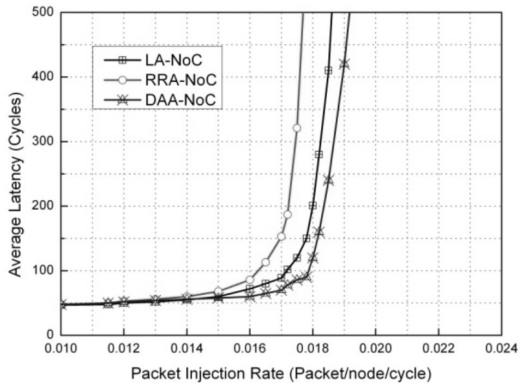
As shown in fig. 6, RRA-NoC, DAA-NoC and LA-NoC perform almost the same when the networks are under light to moderate packet injection rate because there are few full buffers. So, most input ports have an equal chance to occupy the desired output ports. Although some input ports have less chance to win the desired output ports, they have free buffer slots for holding the incoming packets that results in no harm to packet latency. When the networks start to approach saturation, DAA-NoC provides much lower latency compared with RRA-NoC and LA-NoC. For example, when packet injection rate is 0.013 under VOPD pattern, the latencies of RRA-NoC, LA-NoC and DA-NoC are 19 cycles, 18 cycles and 14 cycles respectively. As the packet injection rate is increase to 0.019 under VOPD pattern, the latencies of RRA-NoC, LA-NoC and DA-NoC are 376 cycles, 130 cycles and 91 cycles respectively.

Besides that, The saturation PIR of DAA-NoC also shows an improvement no matter under uniform or unbalanced traffic patterns. For above four traffic patterns, the saturation PIR of DAA-NoC increase by 1.5%, 9%, 7.1% and 11.7% respectively, compared with RRA-NoC. So, the average saturation PIR for DAA-NoC is about 7.3% higher than RRA-NoC. Compared to RRA-NoC, LA-NoC performs worse under uniform traffic pattern, but better under the other three traffic patterns. For uniform traffic pattern, lottery arbiter cannot distinguish the high priority input port clearly. Thus, all input ports win the desired output ports based on luck in LA-NoC. The average saturation PIR of LA-NoC is about 4.2% higher than that of RRA-NoC.

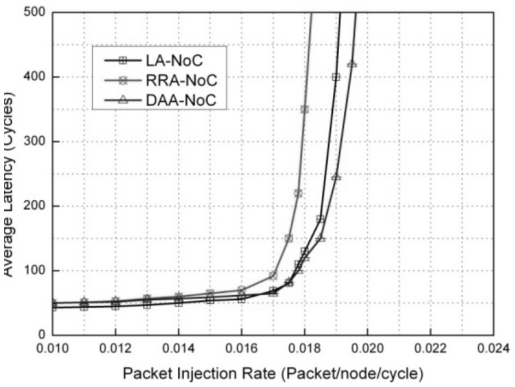
After comparing the latencies and saturation PIRs, fig.7 presents the saturation throughput of different NoCs. Considering the four traffic patterns utilized in this work, the average saturation throughputs of LA-NoC and DAA-NoC increase by 8.5% and 13.3% respectively, compared with RRA-NoC. Even under uniform traffic pattern, the DAA-NoC also achieves about 3.2% saturation throughput improvement compared to RRA-NoC.



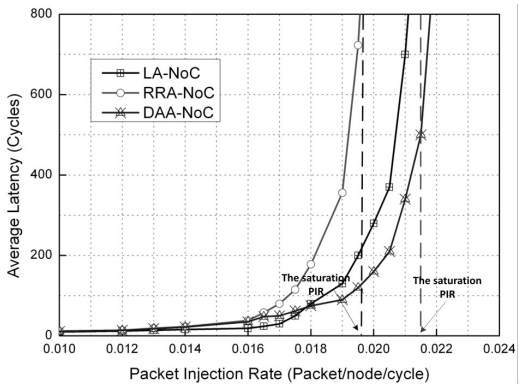
(a)Uniform



(b) Bit-complement



(c) Transpose



(d) VOPD

Figure 6: Average packet latency for NoC with different arbiters

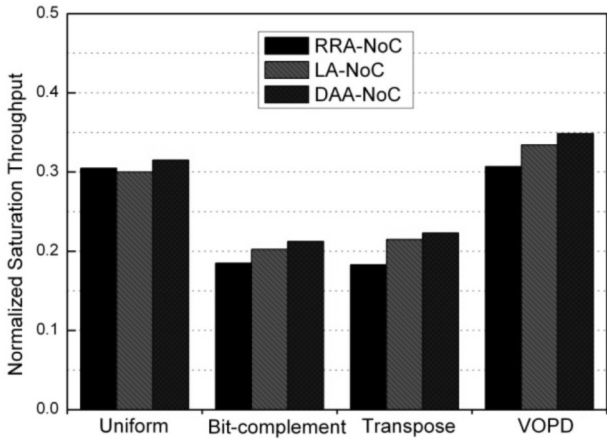


Figure 7: Saturation throughput comparison of NoCs using different arbiters.

4.2 Hardware implementation

Hardware overhead is also an important metric during NoC design. In this section, we first compare the characteristics of round robin arbiter, lottery arbiter and the proposed arbiter. All these arbiters are described in verilog HDL, and synthesized by Synopsys Design Compiler tool with 0.18-μm CMOS library. Table 1 shows the size and the timing results of these designs.

Table 1: Characteristics of different arbiters

	Area (um2)	Size (# of NAND2)	Critical path delay (ns)	Maximum frequency (GHz)
Round robin arbiter	650	72	0.87	1.1
Lottery arbiter	2,735	304	1.38	0.7
Dynamic adaptive arbiter	1,821	202	1.14	0.85

As shown in Table 1, among all the designs, round robin arbiter runs the fastest with its the smallest size. However, as we pointed out and showed before, round robin arbiter is not efficient enough for most communications. Lottery arbiter runs the slowest because of the most complex logic. Compared to round robin arbiter, the size and critical path delay of lottery arbiter increase by 322.2% and 58.6%, respectively. Dynamic adaptive arbiter is better than lottery arbiter, but worse than round robin arbiter. It consumes 180.6% additional hardware resources and causes 31% extra critical path delay compared with round robin arbiter.

Arbiters are small components in a router. Although there are five output arbiters in 2D mesh on-chip router, the area of arbiters makes up only a small portion

of the whole router area. The router using round robin arbiter (RRA-Router), lottery arbiter (LA-Router) and dynamic adaptive arbiter (DAA-Router) are implemented with 200MHz clock frequency. Table 2 gives the detailed area consumptions of these routers.

Table 2: Area overheads for different arbiters in router

design	Area for five arbiters (um2)	Area for the whole router (um2)
RRA-Router	3,250	121,334
LA-Router	13,675	131,762
DAA-Router	9,105	127,192

As table 2 shows, arbiters consume approximately 2.7% to 10.4% of the total area in the different routers. Compared with RRA-Router, LA-Router and DAA-Router require 8.6% and 4.8% additional chip area for the whole router, respectively. However, using DAA-Router instead of RRA-Router, DAA-NoC achieves average 7.3% improvement on saturation PIR and 13.3% improvement on saturation throughput at the cost of additional 4.8% chip area overhead. Thus, the area overhead of the proposed arbiter for on-chip router is acceptable. Compared with LA-Router, DAA-Router pays less area overhead but achieves more performance improvement for NoC. Besides that, the maximum frequencies of these three routers are similar because the critical paths in these routers are determined by the buffer read/write control logics and arbiters have no influence on critical path delay.

5 Conclusions

In this paper, we proposed a dynamic adaptive arbiter based on round robin mechanism. The proposed arbiter detects buffer status of input ports in every clock cycle and adjusts priority of each input port dynamically. It can authorize the input port for transferring data preferentially if the buffer of port is full. Under uniform traffic and non-uniform traffic patterns in NoC, we compared the performance and hardware overhead of NoC based on round robin arbiter, lottery arbiter and the proposed arbiter. The comparison results showed that the proposed arbiter can improve the performance of NoC with an affordable hardware overhead.

Acknowledgments

The research work in this paper is financially supported by National Natural Science Fund (Grant No.61201244), Key Project of Chinese Ministry of Education (Grant No.210080) and the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant No. 12KJA51002).

References

1. L. Benini, G. De Michel, Networks on Chips: A New SoC Paradigm. Computer, Vol. 35, no.1, Jan.2002, pp.70-78.
2. W.J. Dally and B. Towles, Route packets, not wires: on-chip interconnection networks, in Design Automation Conference, 2001. Pro-ceedings , Las Vegas, NV, IEEE Press, 2001, pp. 684-689.
3. P. Guerrier and A. Greiner, A generic architecture for on-chip packet-switched interconnections, in: Design, Automation and Test in Europe Conference and Exhibition 2000. Proceedings, 2000, pp. 250-256.
4. J. Nagle. On packet switches with infinite storage. IEEE Transactions on Communications, vol. 35 no. 4, Apr. 1987, pp.435–438.
5. F. Poletti, D. Bertozzi, L. Benini, A. Bogliolo, Performance Analysis of Arbitration Policies for SoC Communication Architectures, Design Automation for Embedded Systems, Vol. 8, no. 2-3, pp 189-210, 2003.
6. K. Lahiri, A. Raghunathan and G. Lakshminarayana, Lotterybus:A New High-Performance Communication Architecture for System-on-Chip Designs. DAC, Las Vegas, Nevada, USA. 2001. Processings of DAC,200l: 18~22
7. A. Bystrov and A. Yakovlev, Ordered arbiters, Electronics Letters, vol. 35, no. 11, pp. 877-879, 1999.
8. C. Chenghao, T. Kunlin, L. Feipei, T. Shunhung, A Priority based Output Arbiter for NoC Router 2011 IEEE International Symposium on Circuits and Systems (ISCAS), May. 2011, pp. 1928-1931.
9. X.-J. Zhu, H.-B. Zeng, K. Huang, G. Zhang, Round-robin based scheduling algorithms for FIFO IQ switch, IEEE International Conference on Networking, Sensing and Control, 2008, pp. 46-51.
10. Y. Zhang, Architecture and performance comparison of a statistic-based lottery arbiter for shared bus on chip, Proceedings of the Design Automation Conference, Jan. 2005, vol. 2, pp. 1313-1316.
11. C. Wu, H. Li, Y.-B. Li, Z.-M. Yang, Lottery Router: A Customized arbitral priority NoC router. Internal Conference on Computer Science and Software Engineering, Dec. 2008, pp. 411-414.
12. J. Wang, Y.-B. Li, Q.-C. Peng, T.-Q. Tan, A dynamic priority arbiter for Network-on-Chip, IEEE International Symposium on Industrial Embedded Systems, 2009, pp. 253-256.
13. J. Kim et al., A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip

Networks, in Proc. of 33rd International Symposium on Computer Architecture, 2006, pp. 4-15.
14. D. Bertozzi, L. Benini, Xpipes: a network-on-chip architecture for gigascale systems-on-chip, IEEE Circuits and Systems Magazine, vol. 4 no. 2, 2004, pp. 18-31
15. S.-Q. Zheng, M. Yang, Algorithm-Hardware Codesign of Fast Parallel Round-Robin Arbiters, IEEE Transactions on Parallel and Distributed Systems, vol. 18, no. 1, 2007, pp. 84-95.
16. J.M Jou, Y.-L Lee, An Optimal Round-Robin Arbiter Design for NoC, Journal of Information Science and Engineering vol. 26, 2010, pp. 2047-2058.
17. F. Guderian, E. Fischer, M. Winter, G. Fettweis, Fair rate packet arbitration in Network-on-Chip, 2011 IEEE International SoC Conference (SoCC), 2011, pp. 278-283.
18. W. J. Dally and B. Towles, Principles and Practices of Interconnection Networks . San Mateo, CA: Morgan Kaufmann, 2004.
19. M. Janidarmian, A. Khademzadeh, M. Tavanpour, Onyx: a new heuristic bandwidth-constrained mapping of cores onto tile-based Network on Chip, IEICE Electronics Express, Vol. 6 No. 1, 2009, pp.1-7.

Arrived: 03. 01 .2013
Accepted: 08. 05. 2013