Round-trip latency prediction for memory access fairness in mesh-based many-core architectures

Yang Li[1,2], **Xiaowen Chen**[3,4], **Xiaohui Zhao**[1a)], **Yong Yang**[2], **and Hengzhu Liu**[4]

[1] *College of Communication Engineering, Jilin University, China*
[2] *School of Electronics and Information Engineering, Changchun University of Science and Technology, China*
[3] *Department of Electronic Systems, KTH-Royal Institute of Technology, Sweden*
[4] *College of Computer, National University of Defense Technology, China*
a) *xhzhao@jlu.edu.cn*

Abstract: In mesh-based many-core architectures, processor cores and memories reside in different locations (center, corner, edge, etc.), therefore memory accesses behave differently due to their different communication distances. The latency difference leads to unfair memory access and some memory accesses with very high latencies, degrading the system performance. However, improving one memory access's latency can worsen the latency of another since memory accesses contend in the network. Therefore, the goal should focus on memory access fairness through balancing the latencies of memory accesses while ensuring a low average latency. In the paper, we address the goal by proposing to predict the round-trip latencies of memory access related packets and use the predicted round-trip latencies to prioritize the packets. The router supporting fair memory access is designed and its hardware cost is given. Experiments are carried out with a variety of network sizes and packet injection rates and prove that our approach outperforms the classic round-robin arbitration in terms of average latency and LSD[1]. In the experiments, the maximum improvement of the average latency and the LSD are 16% and 48% respectively.

Keywords: round-trip, fair memory access, mesh, many-core
Classification: Integrated circuits

References

[1] S. Borkar: DAC (2007) 746.
[2] J. D. Owens and W. J. Dally: IEEE Micro **27** [5] (2007) 96. DOI:10.1109/MM.2007.4378787
[3] E. Marinissen, B. Prince, D. Keltel-Schulz and Y. Zorian: DATE (2005) 722.

[1]LSD: Latency Standard Deviation measures the amount of variation or dispersion from the average latency. A low standard deviation indicates that the latencies tend to be very close to the mean. Therefore, LSD is suitable to evaluate the memory access fairness.

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014

[4] ITRS 2013 document (2013) www.itrs.net/Links/2013ITRS/Home2013.htm.

[5] P. Pande, C. Grecu, M. Jones, A. Ivanov and R. Saleh: IEEE Trans. Comput. **54** (2005) 1025. DOI:10.1109/TC.2005.134

[6] D. Genius: ReCoSoC (2013) 1. DOI:10.1109/ReCoSoC.2013.6581525

[7] Z. Majo and T. R. Gross: SYSTOR (2011) 1.

[8] M. Daneshtalab, M. Ebrahimi, J. Plosila and H. Tenhunen: DATE (2013) 1048.

[9] W. Jang and D. Pan: IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29** (2010) 1572. DOI:10.1109/TCAD.2010.2061251

[10] T. Pimpalkhute and S. Pasricha: VLSI Design (2014) 234.

[11] A. Sharifi, E. Kultursay, M. Kandemir and C. Das: MICRO (2012) 294. DOI:10.1109/MICRO.2012.35

[12] W. J. Dally: IEEE Trans. Parallel Distrib. Syst. **3** (1992) 194. DOI:10.1109/71.127260

[13] A. Kumar, L. Peh, P. Kundu and N. Jha: ISCA (2007) 150.

1 Introduction, motivation, and related work

It's a trend that high-performance single-chip computing architectures evolves from single-core to multi-cores and even many-cores [1]. In such architectures, Network-on-Chip (NoC) [2] is recognized as the scalable solution to interconnect so many cores and hence has attracted significant attentions over the last ten years since various buses do not scale well with the system size. Besides, on-chip memory content increases from 20% ten years ago to 85% of the chip area today and will go on increasing in the future [3]. Memories are preferably to be distributed for medium and large scale system sizes because centralized memory has already become the bottleneck of performance, power and cost [4]. Fig. 1 shows an example of such many-core architectures with distributed but shared memories. The example system is composed of 25 Processor-Memory (PM) nodes interconnected via a packet-switched 2D mesh network, which is a most popular NoC topology [5] due to its regularity, simplicity and modularity. Each PM node is connected to a router. Routers are interconnected with bidirectional links. A PM node contains a processor core and a local memory that is accessible to all PM nodes. As can be observed, memories are distributed and shared so that the centralized memory organization and hence the hotspot area are avoided.

As shown in Fig. 1, because processor cores and memories are located in different nodes (center, corner, edge, etc.), memories are asymmetric so as to be a kind of NUMA (Non Uniform Memory Access) [6, 7] architecture and memory accesses behave differently due to their different communication distances. For instance, the up left corner PM node accesses the local memory in its neighboring PM node in 2 hops for a round-trip memory read, but it takes 16 hops over the network if it reads a data in the local memory of the bottom right corner PM node. As the network size is scaled up, the communication distance difference of memory accesses becomes bigger and a large number of memory accesses worsen the network contention and congestion, thus the performance (latency) gap of different memory accesses becomes larger. This phenomenon may lead to very high latencies suffered from by some memory accesses that negatively affect the system

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014

performance. To achieve high performance in mesh-based many-core architectures, it is crucial to reduce the number of memory accesses with very high latencies. However, this should be done with care as shortening one memory access's latency can worsen the latency of another because the network resources such as router buffers and communication links are shared. Therefore, we are motivated to focus on balancing the latencies of memory accesses (i.e. narrowing the latency difference of memory accesses) as well as ensuring a low average latency, which is referred to as the "memory access fairness" problem in mesh-based many-core architectures.

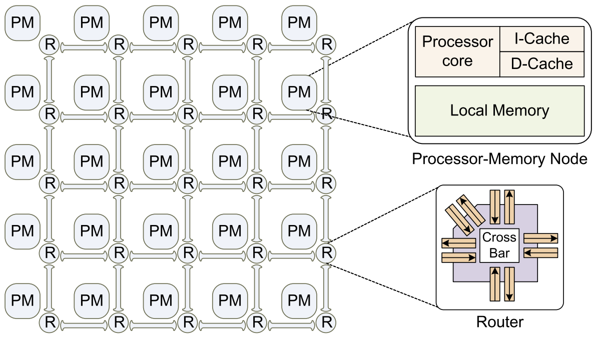

Fig. 1. A 5×5 mesh-based many-core architecture

For memory access fairness in NoC-based multicores, prior work paid attentions to studying fair memory access to the off-chip SDRAM, which is a memory access hotspot region and the latencies of memory accesses may be so different. For instance, in [8], Daneshtalab proposed to prioritize requests in SDRAM memory interfaces according to the congestion information such that requests from less-congested regions prioritize over the other requests. In [9], Jang presented a memory-aware NoC router which performs switch arbitration considering the memory access latency of packets contending for the same output port. The router performs fair memory access scheduling which was previously executed by the SDRAM memory controller. As many memories are integrated and shared in many-core architectures and network latency plays a significant role in on-chip memory access latency, on-chip memory access performance gradually attracts researchers. In [10], Pimpalkhute proposed a holistic solution for intelligently scheduling on-chip and off-chip memory requests to optimize the overall system performance. They balance the latency performance between the on-chip and off-chip requests. Different from them, we focus on balancing the latencies amongst the on-chip memory accesses. In [11], Sharifi addressed balancing latencies of on-chip memory accesses by prioritizing memory response packets such that, in a given period of time,packets that experience higher latencies than the average packet latency of that application are expedited. However, they divided a whole memory access into two parts: outward trip for memory request (read or write) and return trip for memory response (read data or write acknowledgement) and treated them separately. Their scheduling scheme only prioritizes the memory responses and balances the return trip latencies. Different from them, our approach considers the two parts of a memory access as a whole and the round-trip latency is used to prioritize memory

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014

accesses so as to achieve the goal of fair memory access, since considering the entire round trip of a memory access is more reasonable.

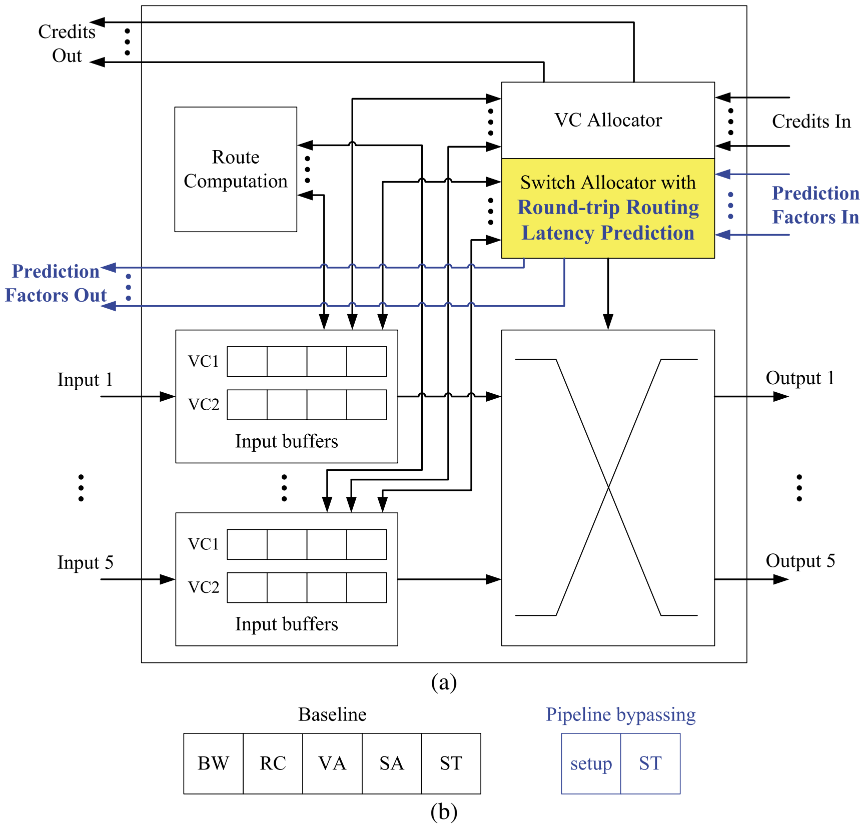

Fig. 2. a) A packet-switched credit-based virtual channel router supporting round-trip latency prediction; b) Pipeline stages in baseline and pipeline bypassing

2 Router design supporting fair memory access

The communication infrastructure in our target architecture is a packet-switched mesh network with deterministic DOR[2] X-Y routing (X first), where packets traveling east or west are allowed to turn south or north but packets traveling south or north are not permitted to turn east and west, thus preventing cyclic dependencies and avoiding network deadlock. Fair memory access is supported in routers during the transmission of memory access packets in mesh-based many-core architectures. In this section, we detail the enhanced router structure with round-trip latency prediction and how the round-trip latency is predicted.

2.1 Router structure

Fig. 2a) shows the structure of our router, which is a state-of-the-art packet-switched credit-based Virtual Channel (VC) [12] router. It is enhanced by supporting round-trip latency prediction implemented in the Switch Allocator (SA) module. Two virtual channels (VC1 for memory request packets and VC2 for memory response packets) are used to break deadlocks induced by the dependency

[2]DOR: Dimension-Ordered Routing

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014

of memory requests and memory responses. The router is designed as a typical structure with 5 logical stage pipeline [13], as shown in Fig. 2b). A packet (its type can be "memory request" or "memory response"), upon arriving at an input port, is first written into the input buffer according to its input VC in the Buffer Write (BW) pipeline stage. In the next stage, the routing logic performs Route Computation (RC) to determine the output port for the packet. The packet then arbitrates for a VC corresponding to its output port in the VC Allocation (VA) stage. Upon successful allocation of a VC, the packet proceeds to the Switch Allocation (SA) stage where it arbitrates for the switch input and output ports. On winning the output port, the packet is then read from the input buffer and proceeds to the Switch Traversal (ST) stage, where it traverses the crossbar to be sent over the physical link finally. To accelerate the packet transmission speed over the router, a mechanism called "Pipeline bypassing" [13] is adopted. The BW, RC, VA and SA stages are combined and performed in the first stage which is named the "setup stage" where the crossbar is set up for packet traversal in the next cycle while simultaneously allocating a free VC corresponding to the desired output port. Therefore, moving one hop takes 1 clock cycle in our NoC. If there is a port conflict in the switch allocation between the packets in the two VCs, the packet with the higher priority is prioritized over the other. To support fair memory access, we propose to use the round-trip latencies of memory accesses as the prioritization base.

2.2 Predicting round-trip latency of a memory access

An entire memory access is a round-trip one containing two parts: memory request (read or write) in outward trip and memory response (read data or write acknowledgement) in return trip, so it behaves as a memory request packet in the first "memory request" phase of its transmission and a memory response packet in the second "memory response" phase of its transmission. The round-trip latency (notated as L) of a memory access can calculated by Formula (1).

$$L = (DL_p + WL_p) + (DL_f + WL_f) \tag{1}$$

The part in the first parentheses is the time that a memory access has consumed during its past transmission, which contains DL_p and WL_p representing the distance latency and the waiting latency[3] in the past transmission respectively. The part in the second parentheses is the time of the left transmission of a memory access, which includes DL_f and WL_f represents the distance latency and the waiting latency in the future transmission respectively. Because our network adopts the deterministic DOR X-Y routing strategy, DL_f is deterministic and Formula (1) is refined as:

$$L = DL_t + WL_p + WL_f \tag{2}$$

where DL_t is the total round-trip distance latency that is equal to DL_p plus DL_f.

To predict the round-trip latency, we need to calculate DL_t, WL_p, and WL_f.

(i) Calculating DL_t

DL_t is known in the deterministic routing network and can be calculated by Formula (3) according to the coordinates of the source and the destination.

[3]Distance latency is the transmission time of a packet without any contention, which is determined by the hop count and the clock cycles per hop. Waiting latency is the time consumed by a packet when it has to wait in the buffer due to its failure of winning the arbitration.

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014

Valid	srcX	srcY	dstX	dstY	WT (Waiting Latency)	VC id	Type	Payload

Type: READ, WRITE, RDATA, WACK

Fig. 3. Packet format

$$DL_t = 2 \cdot (|X_{src} - X_{dst}| + |Y_{src} - Y_{dst}|) \tag{3}$$

where X_{src} and Y_{src} are the coordinates of the source, and X_{dst} and Y_{dst} are the coordinates of the destination. They can be extracted from the packet (see Fig. 3).

(ii) Obtaining WL_p

WL_p is obtained from the "waiting latency (WL)" field in the packet. Fig. 3 illustrates the packet format. when a memory access starts, "WL" is initialized as zero in its memory request packet. The initial value of "WL" in its memory response packet is equal to the "WL" value of its memory request packet at the time when it reaches the destination local memory. "WL" is incremented by 1 per clock cycle when the memory request packet or the memory response packet is blocked in the buffer due to the arbitration failure.

(iii) Estimating WL_f

WL_f represents the possible waiting latency of a memory access in its future transmission. To estimate WL_f, we propose to use the number of the occupied items of the related input buffers in the downstream routers along the left routing path of a memory access. For instance, when a packet (notated as A) is going to its 1st downstream router (notated as R) through the inport whose input buffer has 2 packets (the first and the second are notated as $B1$ and $B2$ respectively), it will have to wait 1 hop (1 cycle in our design) for the departure of packet $B2$ until it passes through router R, since packet $B1$ leaves router R at the same time when packet A enters router R. Therefore, the future waiting time of packet A in its 1st downstream router R is considered to be 2. WL_f is estimated to be the sum of the count of the occupied items of the related input buffers in all downstream routers along the left routing path. Two steps are used to estimate WL_f as follows:

1. The list (notated as $\mathbb{R}$) of the downstream routers in the left routing path is obtained according to the packet's source and destination coordinates.
2. WL_f is calculated by Formula (4).

$$WL_f = \sum_{R \in \mathbb{R}} FWL(R) \tag{4}$$

where $FWL(R)$ is the function of calculating the future waiting latency in the downstream router R and shown in Formula (5).

$$FWL(R) = \begin{cases} \eta_R - \delta_R, & when\ \eta_R > \delta_R \\ 0, & when\ \eta_R \leqslant \delta_R \end{cases} \tag{5}$$

where δ_R is the hop count from the current router to the downstream router R and η_R is the number of the occupied items of the related input buffer in the downstream router R.

The two-step prediction method above considers the potential waiting latency due to the Head-of-Line blocking induced by the packets that has existed in the input buffers in the left routing path. It does not predict the possible waiting latency

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014

```
$P_j = 0;$
for ($i = 1; i <= 5; i{+}{+}$)
  if ($P_j < P_{i,j}$) $P_j = P_{i,j};$
```

Fig. 4. The arbitration policy based on the predicted round-trip latency (L)

due to the resource contention with other packets because the arrival time of other packets is undetermined and contention is hard to be described properly and estimated accurately. The two-step prediction method is generic and can be simplified when in concrete implementation. From Formula (5), we can see that the packet will not wait in the downstream router R if the number of packets in the related input buffer in the downstream router R is less than or equal to the hop count between the current router and the downstream router R. Therefore, the first step can be simplified to get the nearest 3 downstream routers in the left routing path since the depth of the input buffer in our router is 4. In the router design, all of η_R in the nearest 3 routers come together to be the "Prediction Factors In" input and all of η_R in the current router form the "Prediction Factors Out" output, as shown in Fig. 2.

2.3 Prioritization

To achieve the goal of fair memory access, in our router design, the predicted round-trip latency (L) is used as the prioritization base to decide arbitration for the memory access related packets. The arbitration policy is that the packet experiencing the highest L is expedited when multiple packets contend for the same output. The formulated arbitration policy is shown in Fig. 4. In the figure, $P_{i,j}$ denotes the packet from input i to output j, while P_j denotes the packet that wins the output j. Here, $i, j \in \{1, 2, 3, 4, 5\}$. $P_{i,j} = 0$ means that there is no packet from input i to output j.

2.4 Hardware cost

The router design is synthesized in Synopsys® Design Compiler with TSMC® 45 nm process. Table I lists the logic synthesis results excluding the wire cost.

Table I. Logic synthesis results of the router

	Area	Frequency
Combinational Logic (RC+VA+SA+ST)	28145.21 μm^2 (29.92k NAND gates)	1.96 GHz (0.51 ns)
Sequential Logic (Input Buffers)	17532.28 μm^2 (18.64k NAND gates)	

Note: The area of a NAND gate with two inputs is 0.9408 μm^2.

3 Experiments and results

3.1 Experimental setup

To evaluate the fair memory access performance of our approach, a homogenous mesh-based many-core simulator is implemented with Verilog and uniform syn-

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014

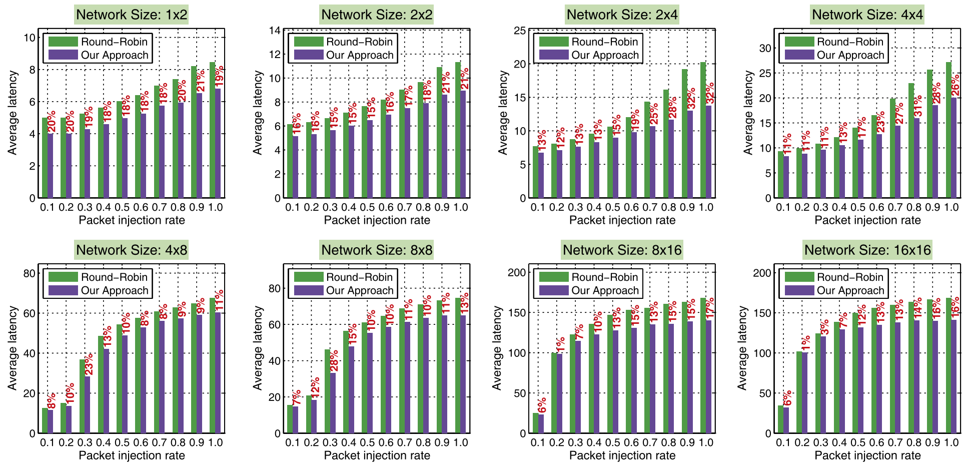

Fig. 5. Comparison of average latency between round-robin arbitration and our approach under different network sizes and packet injection rates

thetic traffic patterns are considered. In experiments, each processor core begins to generate 10,000 memory requests to randomly selected destination local memories when an experiment starts, and an experiment finishes after all processor cores receive their related 10,000 memory responses. All of the experiments are performed with a variety of network sizes and packet injection rates. For performance comparison, we take the classic widely used round-robin arbitration as the counterpart and evaluate average latency ($\bar{L}$) and latency standard deviation (LSD) that are defined by Formula (6) and (7) respectively:

$$\bar{L} = \frac{1}{N}\sum_{i=1}^{N} L_i \tag{6}$$

$$LSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(L_i - \bar{L})^2} \tag{7}$$

where N is the total number of memory accesses and L_i is the round-trip latency of the memory access with the id of i.

3.2 Performance evaluation

Fig. 5 and 6 respectively plot average latency and LSD under different network sizes and packet injection rates. From these figures, we can see that:

- Compared with the classic round-robin arbitration, our approach can has lower average latency and LSD, thus making the latencies of memory accesses more balanced and achieving fairer memory access performance.
- As the network size is scaled up and the packet injection rates increases, our approach can basically gain more performance improvement in terms of average latency and LSD, meaning that, under large-scale network size with a large number of memory accesses, the classic round-robin arbitration has large latency gap of different memory accesses and our approach can balance the latencies of memory accesses well. For instance, under the network size of 16×16 and the packet injection rate of 1.0, in comparison to the round-robin

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014

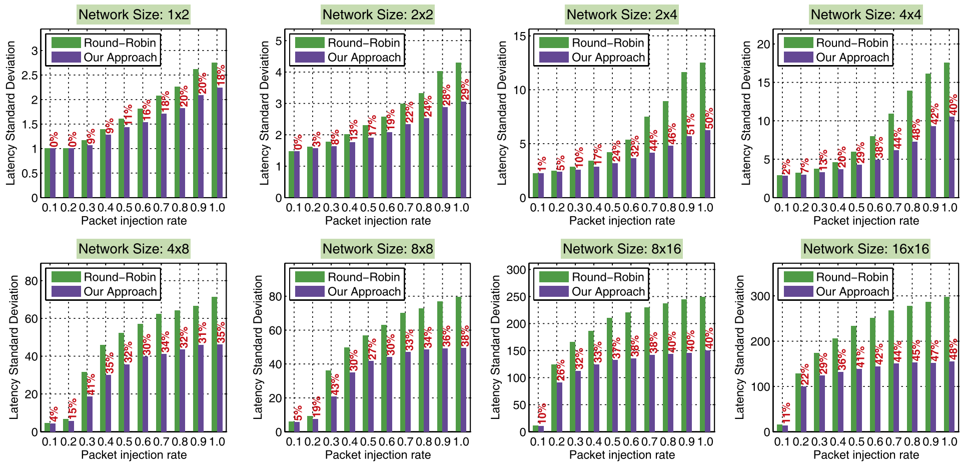

Fig. 6. Comparison of latency standard deviation between round-robin arbitration and our approach under different network sizes and packet injection rates

arbitration, the average latency and the LSD are improved by 16% and 48% respectively, which are the maximum performance improvement we obtain in the experiments.

4 Conclusion

As the number of memory accesses increases largely in mesh-based many-core architectures, the performance (latency) gap of different memory accesses becomes bigger, resulting in some memory accesses with very high latencies and thus negatively affecting the system performance. To achieve the goal of memory access fairness, the paper proposes to predict the round-trip latencies of memory access related packets and use the predicted round-trip latencies to decide arbitration for the packets. The router supporting fair memory access is designed, its hardware cost is reported, and experiments are carried out. The results show that our approach outperforms the classic round-robin arbitration in terms of average latency and LSD, and can achieve the goal of memory access fairness.

In the future, we plan to link and optimize our approach on other irregular and regular large-scale NoC topologies.

Acknowledgment

The research is partially supported by the National Natural Science Foundation of China (No. 61171079) and the Doctoral Program of the Ministry of Education in China (No. 20134307120034).

© IEICE 2014
DOI: 10.1587/elex.11.20141027
Received October 30, 2014
Accepted November 13, 2014
Publicized December 4, 2014
Copyedited December 25, 2014