Novel Self-Compacting Buffer Schemes to Improve Performance of Systems with Network on Chip

Jin Liu[1], Hongmin Ren[1], José G. Delgado-Frias[2], Jeong-Uk Kim[3] and Jin Wang[4]

[1] College of Information Engineering, Shanghai Maritime University, Shanghai, China
[2] School of EECS, Washington State University, Pullman, WA, USA
[3] Department of Energy Grid, Sangmyung University, Seoul 110-743, Korea
[4] Computer and Software School, Nanjing University of Information Science & Technology, Nanjing, China
{jinliu, hmren}@shmtu.edu.cn; jdelgado@eecs.wsu.edu; jukim@smu.ac.kr;
wangjin@nuist.edu.cn

Abstract

In this paper, we present three novel input buffer schemes for systems with network on chip. These proposed buffer schemes are based on a self-compacting buffer, and can provide larger available buffer space per physical channel for communicating applications. These schemes outperform existing approaches. $DAMQ_{shr}$ has similar performance using only sixty three percent of the buffer size that is used in traditional implementation for NoCs. $DAMQ_{min}$ provides an excellent technique to optimize buffer management, and provides a good throughput when the network has a larger load. In addition, they also have better utilization of the available buffer space.

Keywords:Self-compacting, buffer, DAMQ , Network on Chip

1. Introduction

As technology allows greater integration system-on-chip (SoC) designs have a great potential. SoCs are anticipated to have a number of components (or modules) that will interact to compute a solution. These components will need to communicate to transfer data and/or control information. Having dedicated connections between any given modules could be extremely complex as the number of modules increases. An alternative could be use an interconnection network within the chip. An interconnection network on chip needs be restricted in terms of area due to the constraints of being in a single chip. Thus, it is extremely important to use schemes that require less hardware resources as well as provide a good performance. In this paper we present three novel buffer schemes that are based on a dynamically allocated multiple queue (DAMQ) Self Compacting buffer. While providing better performance than traditional statically allocated multiple-queue (SAMQ) and DAMQ buffers, they use less buffer space and, therefore, requiring less hardware.

This paper has been organized as follows. In Section 2, background information is provided. In Section 3, the proposed buffer schemes are described. Performance evaluation results are reported in Section 4. Some concluding remarks are included in Section 5.

2. Background

Routing and buffer organization algorithms are two important design factors of a modern high performance wormhole switch for an interconnection network on chip. In this section we present a brief review of existing routing algorithms and DAMQ buffer schemes which is widely accepted as an efficient way to organize a switch's input buffer.

Routing algorithms are used to determine the messages path from source to destination. They can be implemented in two ways which are deterministic and adaptive. Deterministic routing protocol chooses the path for a message only by its source and destination. The packet will be delayed if any channel along this path is loaded with heavy traffic, and if a channel along this path is faulty the packet cannot be delivered. Thus the deterministic routing protocols are prone to suffer from poor use of bandwidth [1]. Adaptive routing protocols are proposed to make more efficient use of bandwidth and to improve fault tolerance of interconnection network. In order to achieve this, adaptive routing protocols provide alternative paths for communicating nodes. Thus it can overcome the congested areas in the network. Several adaptive routing algorithms have been proposed, showing that message blocking can be considerably reduced, thus strongly improving throughput [2]. Routing algorithms based on Duato's design methodology [3] are widely used. These routing algorithms split each physical channel into two virtual channel sets, the adaptive and the deterministic channels. Packets are routed adaptively using any available adaptive channel or the deterministic channels which form escape paths. When the paths of adaptive channels are blocked, a packet uses an escape channel at the congested node. In this study, we choose one of them to conduct simulations.

DAMQ input buffer had been shown to have better performance than SAMQ buffer. There are several methods to implement DAMQ input buffer in the literature. The first way is to use a linked list scheme [4, 5]. The basic idea of this approach is to maintain (k+1) linked lists in each buffer: one list of packets for each one of the (k-1) output ports, one list of packets for the end node interface and one list of free buffer blocks. Corresponding to each linked list there is a head register and a tail register. The head register points to the first block in the queue and the tail register points to the last block. In each output queue, next block information also must be stored in each buffer block to maintain the FIFO ordering. The second way is to adopt self-compacting buffer scheme. Self-compacting buffer (SCB) was proposed to reduce the hardware complexity incurred by the linked list scheme [6]. The idea is to divide the buffer dynamically into regions with every region containing the data associated with a single output channel. If two channels are denoted as i, j with $i < j$, then the addresses of buffer regions for the two channels $A_i A_j$ will be $A_i < A_j$. There is no reserved space dedicated for any channel. Data is stored in a FIFO manner within the region for each channel. When an insertion of the packet requires space in the middle of the buffer, the required space will be allocated by moving down all the data which reside below the insertion address. Similarly, when a read operation is conducted, buffer space that contains the data will be marked as empty, then the data below the read address is shifted up to fill the empty space.

3. The Novel Buffer Schemes

3.1. DAMQ with Reserved Space for all Virtual Channels Scheme

DAMQ dynamically allocate buffer blocks according to the packet received. Compared with statically allocated buffer scheme, the advantage of DAMQ is to efficiently use the buffer by applying free space to any incoming packet regardless its destination output port.

Since there is no reserved space dedicated for each output channel, the packets destined to one specific output port may occupy the whole buffer space thus the packets destined to other output ports have no chance to get into the buffer. This is the case especially for small and compact routers with limited buffer space where wormhole switching technique and virtual channel mechanism are used. When several virtual channels multiplexed across the physical channel and share a buffer, the virtual channels which have packets accepted in the buffer prior to other virtual channels may hold the whole buffer space when the output port to next hop it destines to is busy. In order to overcome this problem, we implement a new buffer scheme, DAMQ with reserved space for all virtual channels (DAMQ$_{all}$). DAMQ$_{all}$ is based on Self-compacting buffer (SCB) scheme. It inherits most features of the SCB, the virtual channels (VCs) multiplexing one transmission direction of a physical channel (PC) share a buffer. The new feature added is that there is reserved space dedicated for each VC, therefore at any time there is free space for the packets of "late" VCs which has not received packet and one VC can never consume the whole buffer. As shown in Figure 1, the reserved spaces for each VC are arranged sequentially according to the VC's sequence numbers. One register is used to point to the head of each reserved space, i.e. the head of the buffer region for each virtual channel. If two channels are denoted as V_i, V_j with $i + 1 = j$, then the reserved region for $\underline{V_j}$ will be placed right after the reserved region for V_i. The size of reserved space for each virtual channel can be adjusted as shown in Figure. 1 (A). As shown in Figure 1(B), the reserved space for each virtual channel is always kept if there is no flit or only one flit in the buffer region for a specific virtual channel.
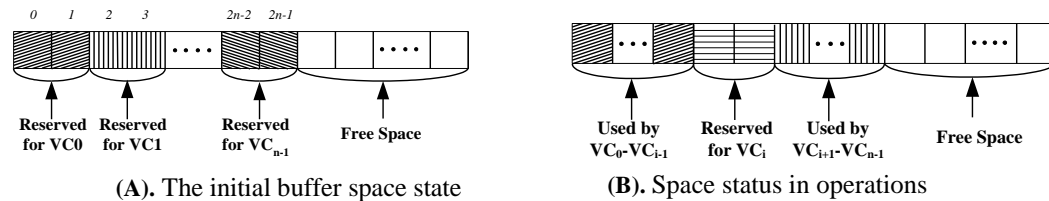

(A). The initial buffer space state   (B). Space status in operations

Figure 1. DAMQ$_{all}$ Buffer Scheme

When the buffer performs shift up or shift down operations, the reserved spaces are treated same as the slots which are holding flits. Thus the order of the buffer space for virtual channels is kept conforming to the sequence of virtual channels. And once the number of current flits in buffer plus the number of reserved slots equals to the total amount of buffer slots, no more flit will be accepted unless this flit belongs to a virtual channel which has any reserved space available. Therefore, one or more virtual channels which have the flits come into the buffer at earlier time can never deprive the chance for other virtual channels which get flits later than them to get buffer. Moreover, once the earlier coming packets are blocked in the buffer, since there is still reserved space for other virtual channels, the network traffic will keep flowing, so the performance of the switch is also enhanced.

3.2. DAMQ with Minimum Reserved Space for Virtual Channels Scheme

DAMQ$_{all}$ improved SCB by reserving buffer space for each virtual channel to avoid the situation that a few VCs consume the whole buffer then other VCs cannot get buffer, even when those VCs which get buffer are blocked. In the simulation experiments, we found that DAMQ$_{all}$ is not the most efficient way to reserve space for VCs. Common cases are there may be no packets destined for some VCs for a while. Even worse situation is that there may be no packets destined for some VCs for a very long time. In either case, the reserved spaces for

these idle VCs are wasted. In order to reserve the buffer space more efficiently and provide more space for flowing traffic, we implement another buffer organization scheme, DAMQ with minimum reserved space for all virtual channels (DAMQ$_{min}$). DAMQ$_{min}$ is also based on SCB scheme and the VCs multiplexing one transmission direction of a PC still share a buffer. The difference to DAMQ$_{all}$ is that at any time there is at most one reserved space for all theVCs. And if every VC have flit present in the buffer, no reserved space will be kept anymore. Thus DAMQ$_{min}$ use minimum space for reserve purpose. As shown in Figure 2(A), buffer slots are reserved for the VC which may firstly claim for buffer:
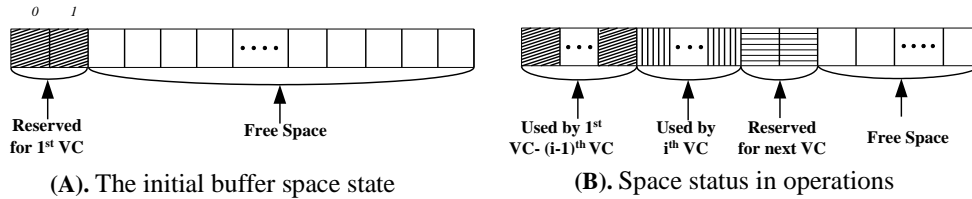

(A). The initial buffer space state　　　　**(B).** Space status in operations

Figure 2. DAMQ$_{min}$ Buffer Scheme

Once a VC has one flit come into the buffer, it will occupy the whole reserved space. As shown in Figure 2(B), this VC will hold at least these reserved buffer slots unless it has no flit left in the buffer. Once every flits of a VC moves out the buffer, the header pointer of this VC will be reset to empty and there are no more buffer slots belong to it. New reserved buffer region may be created if possible. Reserved space is always placed right after the buffer region of a VC which is the latest one to have flit into buffer. When the buffer performs shift up or shift down operations, all reserved slots are treated same as the slots which are holding flits. By dynamically creating reserved space for VCs, DAMQ$_{min}$ presents a very efficient way to use buffer space, there is always minimum buffer space used for reserve purpose, so there are more free space available for flowing traffic.

3.3. Shared DAMQ Buffer Scheme

In a wormhole-switched network with several VCs multiplexing across a PC, routing algorithms tend to choose one set of VCs over others; thus some VCs are more intensively used than others. The traffic load is not evenly distributed in the entire buffer space for a physical channel. Therefore a more efficient approach to use the available buffer space is to let the VCs belonging to a PC share buffer with VCs of another PC to better cooperating with routing methods. To achieve this, our new DAMQ$_{shr}$ buffer design combines the buffer space for VCs from different PC. Based on the statistics of VC usage that are obtained in simulations for Duato's routing algorithms, the pairs with better performance are east X and south Y virtual channels, and the buffer space for west X and north Y likewise. By doing that, one set of X dimension VCs share buffer with one set of VCs from Y dimension. As shown in Figure 3(b), there are two buffers for four physical channels; each buffer is shared by eight virtual channels, and has two read ports and write ports respectively. As shown in Figure 3, space is reserved for each virtual channel sequentially before any flit comes in the buffer. VCs on the X dimension have smaller sequence numbers than those on the Y dimension; the first virtual channel on Y dimension is contiguous to the last one on X dimension. One register is used to point to the head of each reserved space, i.e. the head of the buffer region for each virtual channel.

DAMQ$_{shr}$ operates similarly as DAMQ$_{all}$, except that the number of VCs sharing a buffer is doubled and this is a slightly more control logic overhead.
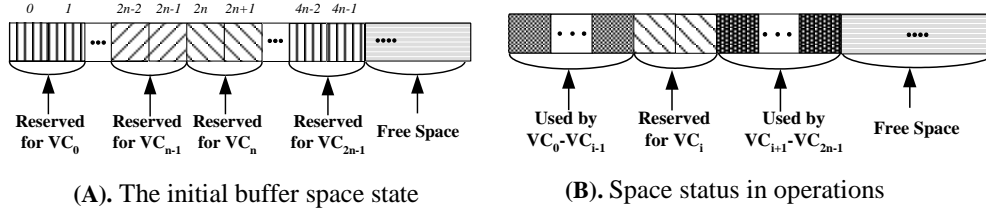
(A). The initial buffer space state   **(B).** Space status in operations

Figure 3. DAMQ$_{shr}$ Buffer Scheme

4. Performance Evaluation

In this section, we describe the results of simulation experiments conducted to evaluate the performance of the proposed buffer organization scheme. Firstly we describe our methodology, configuration of simulation environment and the performance metrics we used. Then we examine the buffer schemes' performance in greater detail.

4.1. Simulated Network Configurations

We have carried out our simulations based on a flit level simulator, flexsim1.2 [7] The architecture we simulate is a 4-ary 2-cube message exchanging system with wrapped around channels. Each switch is attached to one end-node which has one injection channel to the switch and one input channel to receive message from network. Physical channels are duplex channel. Every end node generates packets with randomly determined destination and injects them into the network. Other pertinent simulation configuration parameters are as follows:

- Routing flit delay is set to 1 cycle.

- Data flit delay is set to 1 cycle.

- Buffers at local end nodes for injection are infinite.

- Packets size is set to 32 flits.

- Switching technique used is wormhole.

- The number of virtual channels multiplexing one transmission direction of a duplex physical channel is 4

We use adaptive routing protocol to conduct our simulations as several researchers have reported the strong performance brought by adaptive routing [2]. Duato's routing methodology with dimension order path selection is used in our experiments. We compared the performance of the buffer schemes under uniformly distributed traffic to conform to many other researchers work [8, 9].

4.2. Performance Metrics

We vary the applied load, buffer size; study their impact on throughput and message latency for the network. Since messages are divided into flits when transmitting, to increase message length has the similar effect on increasing traffic load as to shorten the average injection period for each node. We set the message length at 32 flits and make the network into saturation state by shortening injection period for each node to increase the traffic load rate. Traffic load rate is derived by the following equation [7]:

$$LoadRate \times Num_{channel} = Num_{node} \times (MsgLen/IP) \times Dist$$

where *MsgLen* is message length, *IP* is average injection period, and *Dist* is average routing distance. We compare our new buffer schemes with the traditional statically allocated buffer scheme (SAMQ). Traditional DAMQ scheme is not included in comparison, because during the simulation process we found the whole buffer space for one port is easily occupied by the blocked messages which incur deadlock when the network becomes saturated.

Two most important metrics, network throughput and message latency are used to compare these three buffer schemes. The network throughput is defined as the number of flits received per node per cycle as follows:

$$Throughput = \frac{MsgNum \times MsgLen}{Num_{node} \times ST}$$

Where *MsgNum* is the total number of delivered messages; *MsgLen* is the message length, Num_{node} is the total number of nodes and *ST* is the simulation time (in cycles). The message latency is measured as the average time span for every packet between the moment it was generated and the reception of the whole packet at destination. We use average latency ($Latency_{avg}$) of all the injected messages as the performance metric in our simulations. This latency is defined as follows:

$$Latency_{avg} = \frac{1}{Num_{msg}} \times \sum_{i=1}^{Num_{msg}} Latency_i$$

Where $Latency_i$ is latency for Message *i* and Num_{msg} is the total injected message number. In addition, we compared buffer space utilizations among these three buffer schemes to get an in-depth understanding of buffer usage efficiency. We use average stored flits ($Flit_{avg}$) in the buffers of all the nodes as a metric in our simulations. It is defined as following equation:

$$Flit_{avg} = \frac{1}{ST} \times \sum_{1}^{ST} \sum_{i=1}^{Num_{VC}} Flit_i$$

Where $Flit_{avg}$ is the average number of flits that are stored in the buffer space of all the nodes, $Flit_i$ is the number of stored flits in VC's buffer space and Num_{VC} is the total virtual channel count in the network.

4.3. Simulation Results

According to other researchers work in the literature [9, 10], we set the buffer size for each virtual channel to 4 flits. Since four VCs are multiplexing cross one PC, the buffer size for each direction of a duplex physical channel is 16 flits. In order to closely examine the performance of DAMQ$_{shr}$ we use two different size 10 and16 flits buffer. It is well known that there are no significant differences for different buffer schemes while the network has low traffic load. In our simulation experiments, we compare the performance of different buffer schemes when the network starts to saturate on traffic load rate 0.2 until it gets severely saturated after about 0.7 traffic load is applied. As shown in Figure 4, along with the network saturation process, our new DAMQ$_{shr}$ has highest throughput when they all use same size 16-flits buffer. Furthermore, 10-flit DAMQ$_{shr}$ achieves approximately the same maximum throughput as SAMQ using 16-flit buffer. DAMQ$_{min}$ yields slightly better throughput than DAMQ$_{all}$. In sum, DAMQ$_{shr}$ achieves best performance among the four buffer schemes we tested.

As to the message latency, DAMQ$_{shr}$ managed to hold a similar latency as SAMQ until the network is saturated. This is shown in Figure 4. When we further increase the traffic load after the network gets saturated, DAMQ$_{shr}$ and DAMQ$_{min}$ show higher

latency than other schemes. The reason is that they hold much more flits in the buffer than other schemes. Because message latency is a time span average on all the flits that are injected into the network and delivered, more flits stored in the buffer means longer time for them in the waiting queues especially when the network has become drastically saturated. As the total available buffer space is 1024 flits in our simulations, under a traffic load of 0.5, when the simulated buffer schemes use same size 16-flit buffer, $DAMQ_{shared}$ utilizes 56% of the whole buffer space, while $DAMQ_{min}$, $DAMQ_{all}$ and SAMQ uses 46%, 43% and 30% respectively as shown in Figure 6. In addition, 10-flit $DAMQ_{shr}$ contains a similar number of flits as 16-flit SAMQ does.
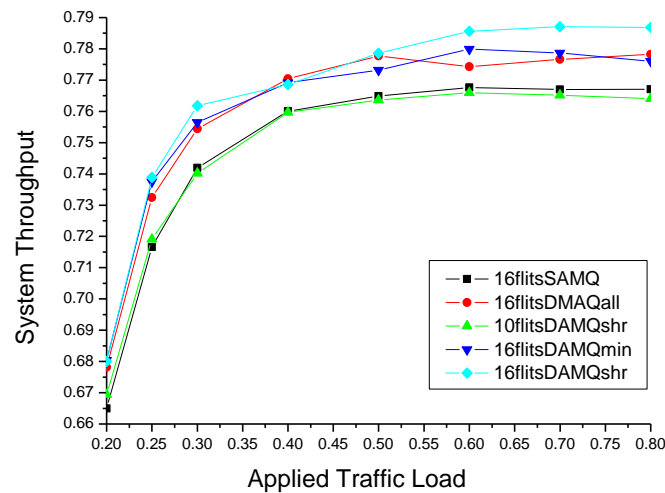

Figure 4. Comparison on Throughput between 16 flit Buffer DAMQ$_{shr}$, DAMQ$_{all}$, DAMQ$_{min}$, SAMQ and 10 flit DAMQ$_{shr}$ under Uniform Traffic
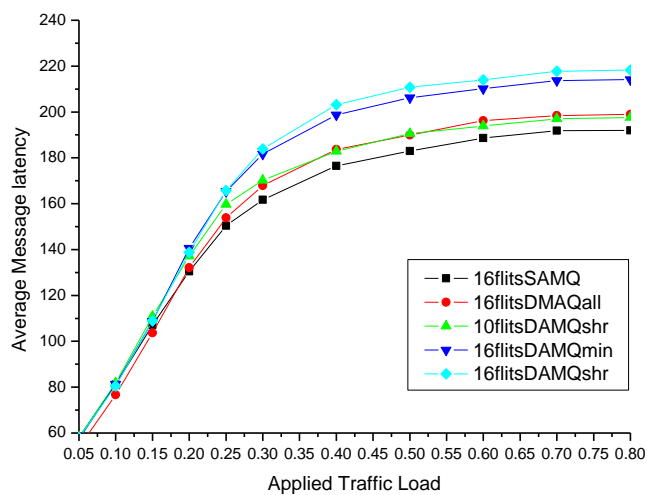

Figure 5. Comparison on Message Latency between 16 flit Buffer DAMQ$_{shr}$, DAMQ$_{all}$, DAMQ$_{min}$, SAMQ and 10 flit DAMQ$_{shr}$ under Uniform Traffic
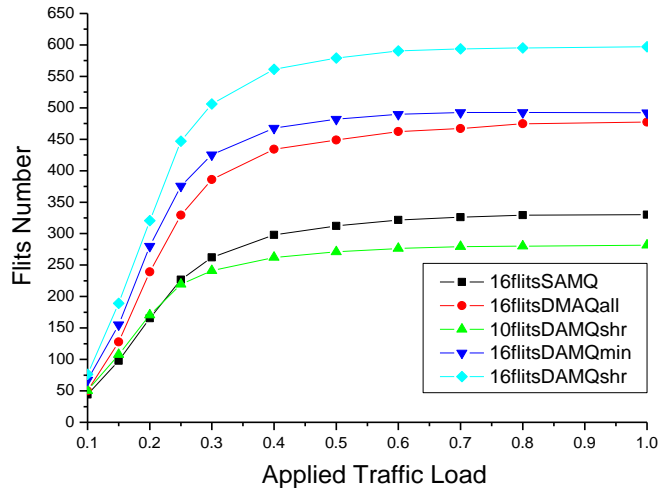
Figure 6. Comparison on Buffer Usage between 16 flit Buffer DAMQ$_{shr}$, DAMQ$_{all}$, DAMQ$_{min}$, SAMQ and 10 flit DAMQ$_{shr}$ under Uniform Traffic

5. Conclusions and Future Work

We have presented three novel buffer schemes based on a DAMQ self-compacting buffer and examined its performance thoroughly. The proposed schemes can all provide similar performance as traditional buffer with significantly less buffer space. If configured with same buffer space, they can provide better system throughput, and DAMQ$_{shr}$ will get the best performance. These buffer schemes also provide better buffer space utilization while DAMQ$_{shr}$ make the best utilization. In summary, when an adaptive routing protocol such as Duato's algorithm is used for the NoC, DAMQ$_{shr}$ is an excellent scheme to optimize buffer management providing a good throughput when the network has a larger load. DAMQ$_{min}$ and DAMQ$_{all}$ can also provide better performance than traditional buffer schemes. Implementing these schemes in hardware requires minor modifications to the self-compacting buffer [11].

In the future work we will keep research on the interaction between buffer organization methods and routing methods and how to make SCB better adapt to different NoC applications.

Acknowledgements

References

[1] P. T. Gaughan and S. Yalamanchili, "Adaptive Routing Protocols for HvDercub Interconnection Networks", IEEE Trans. Computer, vol. 26, no. 5, **(1993)**.

[2] E. Baydal, P. López and J. Duato, "Increasing the Adaptivity of Routing Algorithms for k-ary n-cubes", Proceedings of the 10th Euromicro Workshop on Distributed and Network-based Processing, **(2002)** Jan 9-11, Gran Canaria

[3] J. Duato, "A new theory of deadlock-free adaptive routing in wormhole networks", IEEE Trans. Parallel Distributed Syst., vol. 4, no. 12, **(1993)**.

[4]   Y. Tamir and G. L. Frazier, "Dynamically-allocated multiqueue buffers for VLSI communication switches", IEEE Transactions on Computers, vol. 41, no. 2, (1992).

[5]   R. Sivaram, C. B. Stunkel and D. K. Panda, "HIPIQS: A High Performance switch architecture using input queueing", Proceddings of IPPS/SPDP '98, (1998) March, Orlando, FL.

[6]   J. Park, B. O'Krafka, S. Vassiliadis and J. Delgado-Frias, "Design and evaluation of a DAMQ multiprocessor network with self-compacting buffers", Proceedings of IEEE Supercomputing '94, Conference on High Performance Computing and Communications, (1994) Nov. 14-18,.DC, USA.

[7]   S. Warnakulasuriya and T. M. Pinkston, "Characterization of Deadlocks in k-ary n-cube Networks", IEEE Trans. Parallel and Distributed Systems, vol. 10, no. 9, (1999).

[8]   S. Santi, et. al., "On the Impact of Traffic Statistics on Quality of Service for Networks on Chip", Proceedings of ISCAS 05, (2005) May 27, Kobe, Japan.

[9]   P. P. Pande, et. al., "Performance Evaluation and Design Trade-offs for MP-SoC Interconnect Architectures", IEEE Trans. Computers, vol. 54, no. 8, (2005).

[10] W. J. Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks", Proceedings of DAC 2001, (2001).

[11] J. G. Delgado-Frias and R. Diaz, "A VLSI Self-Compacting Buffer for DAMQ Communication Switches", Proceedings of IEEE Eighth Great Lakes Symp. on VLSI, (1998) February, Lafayette, Louisiana.

[12] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm", IEEE Computer, vol. 35, no. 1, (2002).

Authors

Jin Liu

Dr. Jin Liu received the M.S.degree from University of Electrical Technology of China, Chengdu, China, and the Ph.D degree from Washington State University, Pullman, WA, USA in 2009, all in computer science. He is currently assistant professor at the College of Information Engineering, Shanghai Maritime University, Shanghai, China. He had held industrial R&D position at Windows Core Networking Division, Microsoft, WA, USA. His research interests include wireless sensor network, network on chip, distributed computing and semantic web.

Hongmin Ren

Dr. Hongmin Ren received the M.S.degree from Dalian University of Technology, Dalian, China, the Ph.D degree from Fudan University, Shanghai, and conducted his post doctoral research at Fudan University and Changjiang Computer Group. He is currently associate professor at the College of Information Engineering, Shanghai Maritime University, Shanghai, China. His research interests include software architecture, software components technology and formalization methods..

José G. Delgado-Frias

José G. Delgado-Frias received the BS degree from the National Autonomous University of Mexico, Mexico City, Mexico, the M.S. degree from the National Institute for Astrophysics, Optics and Electronics (INAOE), Puebla, Mexico, and the Ph.D. degree from Texas A&M University, College Station, TX, all in electrical engineering. He is currently Professor at the School of Electrical Engineering and Computer Science, Washington State University,

Pullman, where he holds the Boeing Centennial Chair in Computer Engineering. He has held academic positions at the University of Oxford, State University of New York (SUNY) at Binghamton, University of Virginia, and Princeton University. His research interests include High-Performance VLSI systems, reconfigurable architectures, network routers, and optimization using genetic algorithms.

Jeong-Uk Kim

Jeong-Uk Kim received his B.S. degree in Control and Instrumentation Engineering from Seoul National University in 1987, M.S. and Ph.D. degrees in Electrical Engineering from Korea Advanced Institute of Science and Technology in 1989, and 1993, respectively. He is a professor in SangMyung University in Seoul. His research interests include smart grid demand response, building automation system, and renewable energy.

Jin Wang

Dr. Jin Wang received the B.S. and M.S. degree in the Electronical Engineering from Nanjing University of Posts and Telecommunications, China in 2002 and 2005, respectively. He received Ph.D. degree in the Ubiquitous Computing laboratory from the Computer Engineering Department of Kyung Hee University Korea in 2010. Now, he is a professor in the Computer and Software Institute, Nanjing University of Information Science and technology. His research interests mainly include routing protocol and algorithm design, performance evaluation and optimization for wireless ad hoc and sensor networks. He is a member of the IEEE and ACM.