IPSJ Transactions on Advanced Computing Systems Vol. 4 No. 1 43–52 (Feb. 2011)

Regular Paper

The Implementation of a Low Cost Single-cycle On-chip Router Based on Multiple Virtual Output Queuing

Son Truong Nguyen[†1] and Shigeru Oyanagi[†1]

Network-on-Chip (NoC) is becoming a popular solution for communication on System-on-Chips. A router is a major component of NoC which is responsible for handling the communication. Its architecture significantly impacts on the performance of NoC. In this paper, we propose a low latency router architecture based on virtual output queuing (VOQ). The number of pipeline stages of a packet transfer can be reduced to one stage, by using VOQ buffers and speculatively performing switch allocation and switch traversal in parallel. This paper also proposes a multiple VOQ architecture for which each input port maintains multiple queues for each output channel to improve the throughput of the router. We have implemented the proposed router on FPGA and evaluated in terms of communication latency, throughput and hardware amount. The experimental results show that in a 4×4 two-dimensional mesh network, the proposed multiple VOQ router reduces the communication latency by 25% and cost of area by 15.6% as compared to the look-ahead speculative virtual channel router.

1. Introduction

System-on-Chip (SoC) is an essential design solution for embedded systems with multiple functions, minimized power consumption and energy-efficiency. Generally, the SoC consists of heterogeneous processing elements (such as processors, memories, peripherals, etc... — also called as *IP cores*) and interconnections which are responsible for communication between IP cores. The major challenge of designing such a SoC is a limiting factor for performance, and possibly energy consumption presented by on-chip physical interconnections [1].

In recent years, Network-on-Chip (NoC) is becoming an appealing alternative for communication on SoCs because of its high performance and scalability [2]. A router is a major component of NoC which makes a decision of transmission to route data to the destination according to chosen routing protocols [1),3]. The design and characteristics of a router strongly impact on the total NoC performance.

Typically, at each router a packet transfer is performed by a process of four pipeline stages that consists of the *routing computation* (RC), *virtual channel allocation* (VA), *switch allocation* (SA), and *switch traversal* (ST). This process is one of the main causes of the increased communication latency and may become a critical path of the system.

To reduce the communication latency, various low latency router architectures have been proposed such as *speculation*, *look-ahead*, *bypassing*, and *prediction* [4)–14)]. In these models, the router can bypass or speculatively perform a number of pipeline stages in parallel, determine the route of a packet one hop ahead, or predict an output channel being used by the next packet transfer in order to cut down the critical path.

Numerous existing low latency routers allow to skip one or more pipeline stages speculatively, but they require a complex additional control logic such as the need of two switch allocators [7),8)], path frequency analyzers [11)], preferred path detections [12)], or predictors [13)].

Current commercial NoC routers which have a small number of virtual channels (VCs) may cause the degradation of communication performance on heavy traffic applications. On the other hand, proposals of existing low latency routers may be too complex to be adopted in a SoC environment. The aim of our research is to propose a new router architecture for the SoC which offers a solution of balancing low latency and simplicity of hardware design.

In this paper, we propose a low cost single-cycle router architecture based on the *virtual output queuing* (VOQ) scheme [15)]. In this architecture, each input port maintains a dedicated VC for each output channel (*single* VOQ). By using this simple strategy of VOQ architecture for buffering and speculatively implementing SA and ST operations in parallel, a packet transfer can be performed in only one clock cycle.

In the single VOQ scheme, the traffic congestion at output ports is increased significantly under a heavy network load condition. This leads to the increased queuing latency of the packets and influences the performance of the network. To

†1 Department of Computer Science, Ritsumeikan University

reduce the traffic congestion, a multiple VOQ architecture is proposed. Rather than maintaining a single queue, each input port maintains multiple independent queues for each output channel. The use of multiple VOQ enables to reduce the traffic congestion at output channels, and improve the throughput of the router significantly.

The VOQ technique has been used so far in network switches for avoiding Head-of-Lines (HOLs) [16], however none of the implementations and detailed evaluations has been examined for on-chip routers. In this work, we have implemented the proposed router on FPGA and evaluated in terms of communication latency, throughput and hardware amount. Besides, the VC routers with baseline and look-ahead speculative architectures are also implemented for comparison. The contribution of this paper is to propose a simple low latency router architecture based on multiple VOQ, and to provide detailed evaluations for on-chip routers based on the implementation.

The rest of the paper is organized as follows: In Section 2 we examine the architecture of a conventional VC router, the advantage of the VOQ scheme for input-queued routers, and design alternatives of existing low latency routers. The proposed router architectures based on the single VOQ and multiple VOQ are described in Section 3. The implementation and simulation results are presented and discussed in Section 4 and Section 5 concludes the paper.

2. Background

In this section, we first examine the architecture of a conventional VC router that is considered as our baseline design. Then we discuss the discipline of VOQ and its advantage for input-queued routers. Finally, we discuss the designs of low latency routers developed in recent years.

2.1 Baseline Architecture

In this paper, we assume a dimension-order wormhole router used for the two-dimensional mesh topology as the baseline router, though the router architecture presented readily extends to other topologies. The router has five bi-directional ports named as North (N), South (S), East (E), West (W), and Local (L) for communicating with the neighboring routers and its IP core. **Figure 1** shows the architecture of a baseline VC router [5]. The major components of the router include the input buffers, a routing computation logic, a VC allocator, a switch allocator and a crossbar switch.

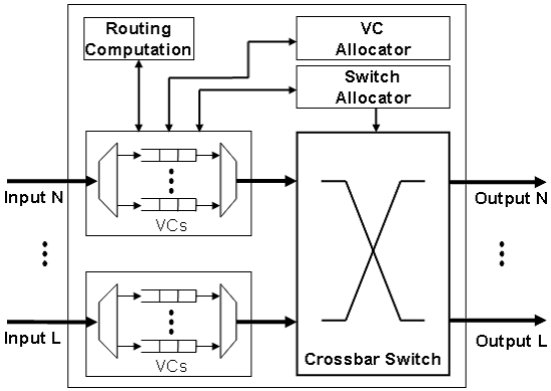

Fig. 1 Architecture of a baseline VC router.

In this model, a packet is generally divided into multiple flits (flow control digits) where the head flit contains all the necessary routing information and the following flits carry payload data. A head flit will advance to the output channel through four pipeline stages that consist of the RC and VA for determining the output channels, the SA for allocating the time slot on the crossbar switch and output channels, and the ST for transferring flits through the crossbar. Once the head flit completes the computation of a route and the allocation of a VC, there is nothing to do for remaining flits in the RC and VA stages. However, they cannot bypass these stages and advance directly to the SA stage because they must remain in order and behind the head flit.

2.2 Virtual Output Queuing Scheme

In a router, the technique of buffering blocked packets is known as the queuing mechanism. One of the queuing strategies used in NoC is *input queuing*. In an input-queued router, packets are buffered at the input ports. Each input port has one or more separate buffers. If a single FIFO (First In First Out) queue is used in an input port to queue receiving packets, then the HOL blocking can occur. HOL blocking is the phenomenon in which the first packet of a queue, that cannot advance to the destination, will block all the other packets queuing

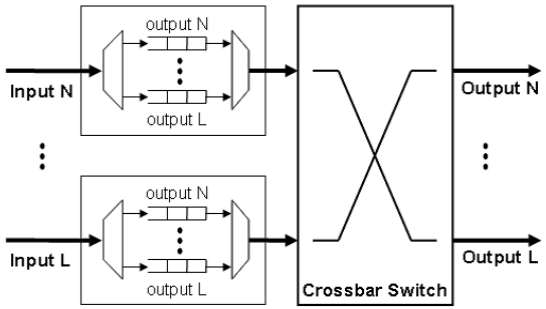

Fig. 2 Virtual output queuing scheme.

behind it. This problem is the cause of increased communication latency and can be avoided by employing VOQ [15),16)].

In the VOQ scheme, each input port of a router maintains a separate queue for each output channel as shown in **Fig. 2**. A packet cannot be blocked by a packet queued ahead of it that is destined for a different output channel, because there is a queue for every output channel.

2.3 Related Works on Low Latency Routers

A well-known low latency router is *speculation* [4)–6)] that speculatively performs several pipeline stages in parallel. Typically, the SA is performed in parallel with the VA (denoted as VSA). Once the speculation succeeds, flits directly enter the ST stage. If the speculation fails, the pipeline stalls and both the SA and VA are repeated on the next cycle. The speculation works well when the router is lightly loaded. However, when the router becomes heavily loaded, it becomes inefficient because of the increased failure of speculation.

The *look-ahead routing* is another technique that reduces the number of pipeline stages by performing the RC ahead of time to remove it from the critical path [5),7),8)]. In this mechanism, each router performs the RC for the next hop (denoted as NRC), and passes the result along with the head flit. Since the control dependency between the RC and others is removed, the NRC and VSA can be performed in parallel to attain a two-stage pipeline. However, performing numerous pipeline stages in parallel may lead to the degradation of frequency and the need of complex control logic.

The *express virtual channels* (EVCs) has been proposed to reduce the communication latency by allowing packets to bypass nodes along their path in a non-speculative fashion [10)]. This method works well for long distance packet transfers in which multiple nodes can be bypassed in large-size networks. However, it is not efficient for neighboring communications and incurs the additional area overhead because of the use of EVCs in addition to the normal VCs.

A *dynamic priority-based fast path* architecture reduces the communication latency by dynamically detecting frequent communication patterns in the network for attaining the fast path, in which the packet transfers can bypass several pipeline stages [11)]. The priority of the fast paths is adjusted during the SA. In this architecture, the use of a path frequency analyzer for determining which paths are used frequently, and a priority-based arbiter for the SA may lead to the increase of hardware cost.

Preferred path is an alternative that enables to achieve low communication latency by defining pre-configured preferred paths adapting the "mad-postman" [12)]. The router uses the preferred path for speculatively forwarding flits to their input's preferred outputs without using a crossbar switch, in addition to the original data path that uses the crossbar. The preferred paths can only be used to reduce the communication latency between the specific source-destination pairs, and require an additional configuration logic for storing and updating their configurations.

Another technique that was introduced by Matsutani [13)] is *prediction*. In the prediction router, the packet transfers can bypass a number of pipeline stages depending on the result of predicting output channels being used by the next packet transfers. If the prediction hits, incoming packets are transferred in a bypass fashion without performing the RC and SA. In case of the failure of prediction, the packets advance to the destination through the original pipeline stages. The difficulty of this scheme is that the applicable prediction algorithms must be investigated for different network environments of topologies, routing algorithms, and traffic patterns. In addition, maintaining a predictor for each VC may cause the complexity of hardware design.

Path-sensitive architecture [14)] is a technique that is based on the utilization of look-ahead routing and speculative strategy with a pre-selection for the output

path. In this model, the actual switch traversal is performed in two stages including the switch traversal to path candidates (ST1) and the crossbar switch traversal (ST2). All the RC, VA, SA, PS (pre-selection) and ST1 are performed concurrently in a speculative fashion. This architecture uses a 4×4 decomposed crossbar instead of a 5×5 full crossbar. Although this decomposed crossbar is useful for neighbor traffics, its impact is trivial for medium and long distance packet transfers. In addition, performing the pre-selection to determine the best path among possible candidates for incoming packets may lead to the complexity of hardware design. Too many operations are speculatively performed in parallel which may also degrade the frequency.

3. Proposed Router Architecture

To reduce the communication latency with the modest area overhead in hardware design, we propose a low latency router architecture utilizing VOQ that helps to reduce the number of pipeline stages of a packet transfer and simplify the hardware micro-architecture.

3.1 Single VOQ Architecture

Our proposed architecture based on single VOQ is illustrated in **Fig. 3**. Each input port maintains a dedicated VC for each output channel. When a packet arrives at an input port, it is stored into the appropriate VC that is determined by its VC identification. At this time, the VC identification field of the packet is updated with a new value that will be used in the next router (next VCID). The next VCID is calculated from routing information that is stored in the head flit of the packet.

Since each input VC is reserved for an output channel, the output port of the packet is easily determined by the current VCID. In addition, the output VC for the packet can be identified by the next VCID (as indicated in Fig. 3). As a result, the pipeline stages of a packet transfer can be reduced to two stages (SA and ST). Furthermore, the number of pipeline stages will be reduced to only one stage if these two stages are speculatively performed in a parallel fashion. In this manner, a single-cycle packet transfer can be achieved. **Figure 4** shows the comparison of pipeline stages of a packet transfer between our proposed VOQ router, the baseline router, and the look-ahead speculative router (with all stages

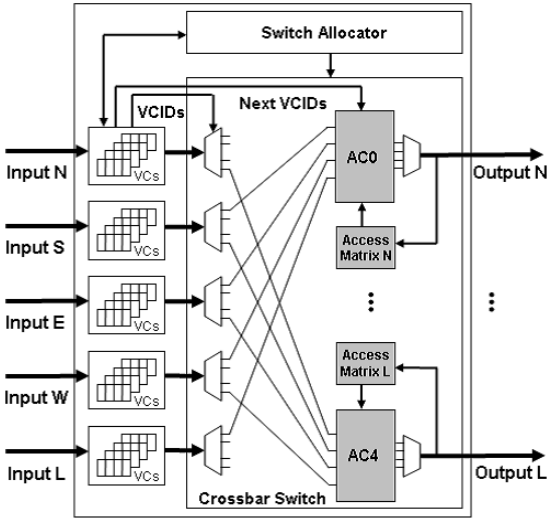

Fig. 3 Single VOQ architecture.

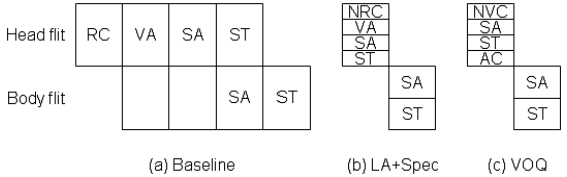

Fig. 4 Comparison of pipeline stages of a packet transfer between (a) baseline router, (b) look-ahead speculative (LA+Spec) router, and (c) proposed VOQ router.

performed in parallel). In our design, since a packet transfer is performed in only one clock cycle, the next VCID (NVC) calculation is performed at the cycle of the head flit arrival, to ensure that in the next cycle the head flit can be forwarded to the output channel with a proper VCID update. Besides, an access control (AC) is performed instead of a VA to guarantee the proper transfer of the packets that are destined to the same output VC (the detail of AC is described later).

At the input port, an incoming flit is first identified by its type. If the head flit of a new packet is detected, the appropriate VC (where it will be stored) is

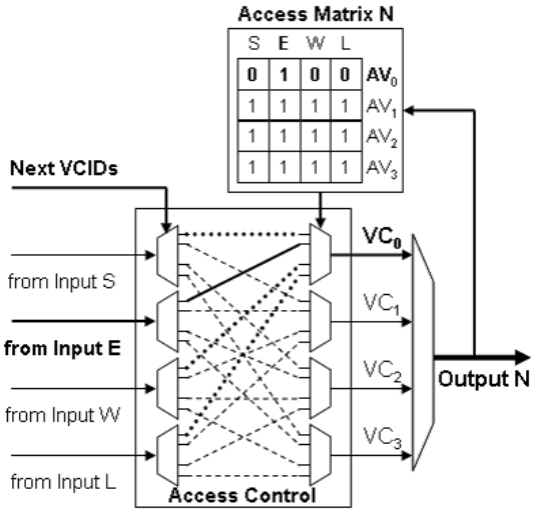

Fig. 5 Access control logic for output VCs.

determined by decoding the VC identification of the flit. The remaining flits of the packet will be stored in the same VC as the head flit. As soon as a flit reaches the head of a VC, a request is sent to the switch allocator to get acceptance for forwarding the flit to the desired output channel. In our design, the VCs are implemented as FIFO queues. To avoid overruns, we have adopted a simple on/off flow control mechanism which simplifies the logic needed to maintain the buffer state.

At the output channel, it is possible for several incoming packets to advance to the same output VC. To ensure that packets are properly transferred in sequence, each output channel maintains an *access matrix* for access control (shown as AC in Fig. 3) to output VCs as illustrated in **Fig. 5**. Each access matrix maintains a separate access vector (denoted as AV in Fig. 5) for each output VC. The access vector is a 4-bit mask used for controlling the access from input channels to the output VC. The masked bit of "1" shows that the input channel can access the output VC, and "0" indicates that the output VC is unavailable for that input channel.

When the head flit of a packet from an input channel advances along an output VC, the output VC is required to be held for all remaining flits of the packet. It is set to be inaccessible for all the other packets of input channels, by updating an appropriate value for its access vector. This state is kept until the tail flit of the packet is forwarded. When the tail flit is detected on an output VC, the state of the output VC is updated as available for all input channels. For example, at the initial time the access matrix is set to be accessible for all incoming packets of input ports (i.e., each AV is set to "1111"). Incoming packets from four input ports will compete for an output VC through the switch allocation. When the input port E takes a path to the VC_0 of output port N, at the instant a head flit is identified on the output VC_0, the AV_0 of the access matrix N is immediately updated to "0100" as available for only input port E. When a tail flit is presented on the output VC_0, the AV_0 of access matrix N is returned to "1111" as available for any input port.

3.2 Multiple VOQ Architecture

In practice, a VOQ router with a single queue for each output channel works extremely well in a light network load. In this situation, the flits of a packet are almost successfully allocated a time slot of the crossbar to the output channel every clock cycle. When the router becomes heavily loaded, the number of packets that contend for the same output VC will increase rapidly. This causes a high traffic congestion status at output ports and leads to the increased queuing latency of packets that degrades the performance of the network.

To reduce the traffic congestion and avoid deadlocks under a heavy network load, we propose a multiple VOQ architecture as shown in **Fig. 6**. Rather than maintaining a single queue, each input port maintains two independent queues called inner virtual channels (inner VCs) for each output channel. These inner VCs share the bandwidth of a single physical VOQ channel. The flow control is performed at two levels: the packet level to assign an inner VC and the flit level to assign a physical channel bandwidth.

At the output channel, incoming packets from eight input inner VCs will compete two output inner VCs as shown in **Fig. 7**. Therefore, each access matrix consists of eight access vectors of 8-bit mask. The winner in the switch arbitration of competitive packets will occupy an output inner VC and make it unavailable for all the other packets. The output inner VC will be free after the tail flit of

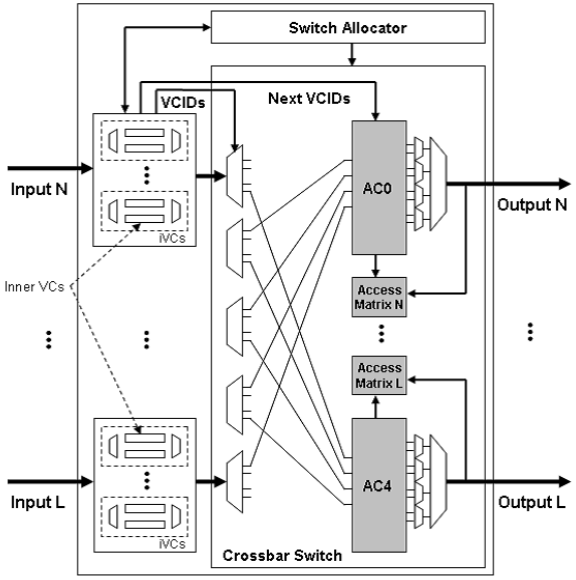

Fig. 6 Multiple VOQ architecture.

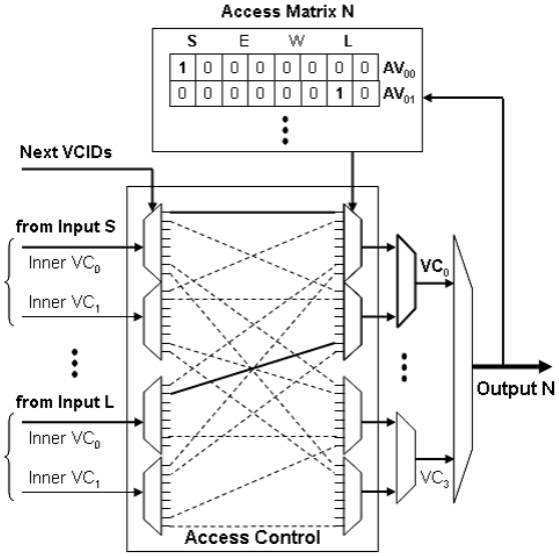

Fig. 7 Access control logic for output inner VCs.

a packet is transferred. For example, in Fig. 7 packets from inner VC_0 of input channels S and L win the switch arbitration and advance to corresponding inner VCs of output VC_0. While these packets advance along their output inner VCs, packets from the remaining inner VCs of input ports cannot be forwarded to the output VC_0 of output port N.

By this way, the use of multiple VOQ enables to reduce the traffic congestion at output ports, and improve the throughput of the router when the network becomes heavily loaded. This helps to reduce the queuing latency of packets, thereby the overall communication latency of the network.

4. Implementation and Evaluation

The router core is implemented in five architectures — the first one with the wormhole architecture without VC, the second one with the baseline architecture, the third one using the look-ahead speculative solution (LA+Spec) [5] in which NRC, VSA and ST stages are performed in parallel, the fourth one with the architecture of a single VOQ, and the last one with the multiple VOQ architecture. All routers have the same parameters as five bi-directional ports, 16-bit data width, and 4-flit buffer size (a 16-flit buffer for the wormhole design and two 2-flit inner VCs for the multiple VOQ architecture).

Our target FPGA device is the Xilinx Virtex-5 XC5VFX70T with 11,200 slices and 148 blocks of BlockRAM. We have used Verilog-HDL for the circuit design, and ModelSim XE III 6.3c from Mentor Graphics Inc. for the functional and structural simulation. The Xilinx integrated tool environment ISE 10.1i is used for the automated logic synthesis, mapping, placing and routing of circuits. Tools included in this environment generate reports describing the area and speed of implementation, a netlist used for timing simulation, and a bitstream used to configure an actual FPGA device.

4.1 Implementation Results

The implementation results of our designs are evaluated in most of the essential characteristics of NoC as shown in **Table 1**. As a result, the router based on

single VOQ scheme can operate at a maximum frequency of 130 MHz, that is decreased by 39.5% and 18.2% in comparison with the wormhole and baseline routers respectively, and increased by 34% as compared to the LA+Spec router. In terms of area overhead (not including BlockRAMs used for buffers), the single VOQ router consumes only 275 slices that is reduced by 65.4% and 67.3% as compared to the baseline and LA+Spec designs, respectively. It is increased by 48.4% as compared to the architecture with no VCs (wormhole router). The single VOQ design achieves a higher speed than the LA+Spec design because of fewer pipeline stages performed in parallel, whereas it is slower than the wormhole and baseline designs which have no pipeline stage performed in parallel.

The breakdown of hardware amount in IU (input unit), RC (routing computation), VA (VC allocation), SA (switch allocation), CB (crossbar), and AC (access control) is presented in **Table 2**. The results show that, in the single VOQ router the SA takes a slightly larger amount of hardware as compared to the wormhole and baseline routers, because the SA and ST stages are performed in parallel.

Table 1 Implementation results of routers.

	Wormhole	Baseline	LA+Spec	SingleVOQ	MultipleVOQ
Number of ports	5	5	5	5	5
Number of VCs	1	4	4	4	4×2
Data width	16-bit	16-bit	16-bit	16-bit	16-bit
Buffer's size	16-flit	4-flit	4-flit	4-flit	2-flit
Slices	142	794	842	275	711
LUTs	535	3,055	3,279	1,060	2,459
Flip-Flops	130	510	820	270	765
BlockRAM	3 blocks	10 blocks	10 blocks	10 blocks	10 blocks
Frequency	215 MHz	159 MHz	97 MHz	130 MHz	102 MHz

However, the IU consumes quite a small number of slices as compared to that in the baseline and LA+Spec routers. It is greater than the wormhole design since it is required to manage a larger amount of buffers. Besides, using the AC instead of the RC and VA requires only a small area overhead that is 46 slices. This reduction of hardware is due to simplifying the RC and VA operations by the use of VOQ.

The design with multiple VOQ architecture has a frequency lower than that of the single VOQ architecture, because the access control to output inner VCs is more complex, but faster than the LA+Spec router and still maintains the cost of hardware resource smaller than the design based on baseline architecture. The reduction of hardware resource in the multiple VOQ design is 10.5% and 15.6% as compared to the baseline and LA+Spec routers, respectively. In comparison with the single VOQ architecture, the multiple VOQ design consumes more hardware resources, because there is a need of control logic for access to the shared inner VCs.

Obviously, the architectures of router based on single and multiple VOQ bring the simplicity of hardware design with significant reduction of hardware resource. In addition, the delay of a packet transfer at each router can be shortened to only one clock cycle.

4.2 Simulation and Observation

Experiments are conducted to evaluate the performance of the routers based on single and multiple VOQ and give a comparison between proposed architectures and the conventional VC architecture. Simulations are carried out on a 4×4 two-dimensional mesh network developed in Verilog. The simulator generates

Table 2 Breakdown of hardware amount.

	Wormhole			Baseline			LA+Spec			SingleVOQ			MultipleVOQ		
	Slices	LUTs	FFs	Slices	LUTs	FFs	Slices	LUTs	FFs	Slices	LUTs	FFs	Slices	LUTs	FFs
IU	69	274	85	456	1,763	215	563	1,971	540	197	644	180	452	1,424	295
RC	-	-	-	60	155	-	60	155	-	-	-	-	-	-	-
VA	-	-	-	231	868	260	231	868	260	-	-	-	-	-	-
SA	11	37	20	11	28	10	16	28	10	17	54	10	17	54	10
CB	62	195	-	71	210	-	70	205	-	70	205	-	70	205	-
AC	14	30	20	-	-	-	-	-	-	46	130	80	234	807	460
Total	156	536	125	829	3,024	485	940	3,227	810	330	1,033	270	773	2,490	765

uniformly distributed traffic across the network to random destinations. Packets are generated at a constant rate and queued until they are able to enter the network. Latency of a packet is measured from the time when the first flit is injected into the packet source queue, to the time when its last flit is ejected from the network, assuming immediate ejection. Each simulation consists of two phases: a warm-up phase of 2,000 packet injections, followed by the main phase which injects 10,000 additional packets. It is assumed that the packet length is fixed to 5-flit.

To minimize the impact of possibly good circumstances in one simulation run, we have conducted multiple runs for each simulation setup and calculated the average network latency from received results of different measurements. The simulation terminates when all packets are received at the destination nodes under normal conditions. Otherwise, the simulation terminates if the average latency of the packets exceeds a threshold of 100 cycles. All nodes inject their flits at the same time for the worst case of a high network load.

In this work, all simulations are performed under a uniform traffic pattern. The communication latency is shown as a function of the injection rate. **Figure 8** shows the network latencies of our proposed routers, the wormhole, baseline and LA+Spec routers with the buffer size of 4-flit (16-flit for the wormhole router). The figure shows that the zero-load latencies of the wormhole, baseline and LA+Spec router are 18, 22 and 16 cycles respectively, while it is only 12 cycles in our proposed single or multiple VOQ router. This 33.3% and 45.5% improvement in comparison with the wormhole and baseline routers is due to the reduction of pipeline stages from three and four to only one cycle, respectively. The latency of our proposed routers is 25% lower as compared to the latency of the LA+Spec router, because in our design only two stages (SA and ST) are performed in a speculative fashion whereas three stages (VA, SA and ST) are performed speculatively in the LA+Spec design. In our work, the implementation of the LA+Spec router is performed without using the pre-computation of grant-enable signals for the VC allocation that is employed in the references [7),8)]. In practice, implementing the pre-computation of grant-enable signals may make the hardware architecture more complex, thereby requiring an additional control logic and may cause the degradation of frequency. Furthermore, when the buffer

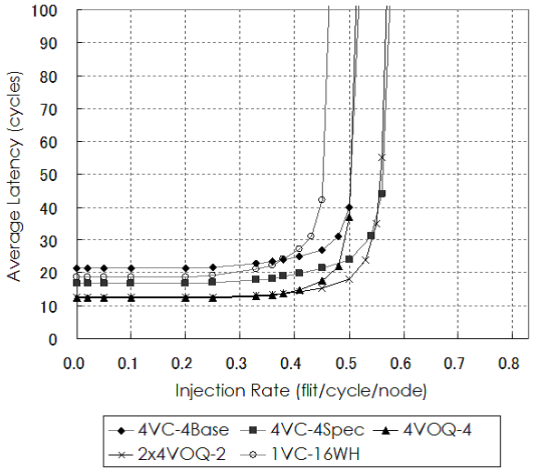

Fig. 8 Network latencies of routers with 4-flit buffer (4VC-4Base = four 4-flit VCs/port Baseline; 4VC-4Spec = four 4-flit VCs/port LA+Spec; 4VOQ-4 = four 4-flit VOQs/port; 2x4VOQ-2 = eight 2-flit VOQs/port; 1VC-16WH = one 16-flit VC/port).

state of multiple output VCs assigned to VCs at a single input port is updated in a single cycle, it is difficult to set all grant-enable signals safely. Therefore, the pre-computation of grant-enable signals is not employed for avoiding the limitation of contention, and simpler hardware design as well as reduced area overhead that may be more suitable for low cost applications on SoCs.

The throughput of our proposed single VOQ router saturates at about 51% of capacity that is similar to the throughput of the baseline router, and 5% higher than that of the wormhole router. In the multiple VOQ architecture, the saturation point of throughput is extended to 56% of capacity. This result is equivalent to that of the LA+Spec router. **Figures 9** and **10** show the results for different sizes of a buffer which are greater than the packet size.

These experimental results show that, the router designs based on the VOQ scheme not only bring the simplicity of hardware design, but also outperform the conventional VC router in terms of communication latency.

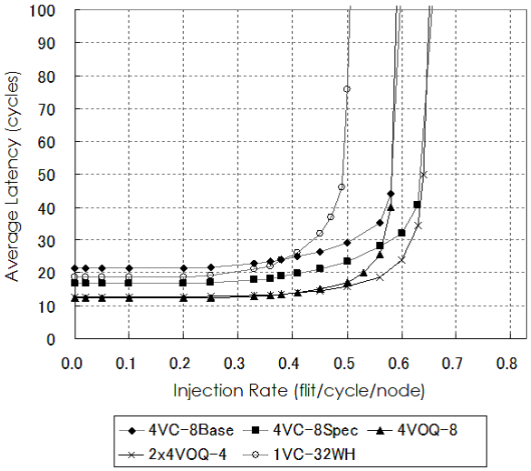

Fig. 9 Network latencies of routers with 8-flit buffer (4VC-8Base = four 8-flit VCs/port Baseline; 4VC-8Spec = four 8-flit VCs/port LA+Spec; 4VOQ-8 = four 8-flit VOQs/port; 2x4VOQ-4 = eight 4-flit VOQs/port; 1VC-32WH = one 32-flit VC/port).

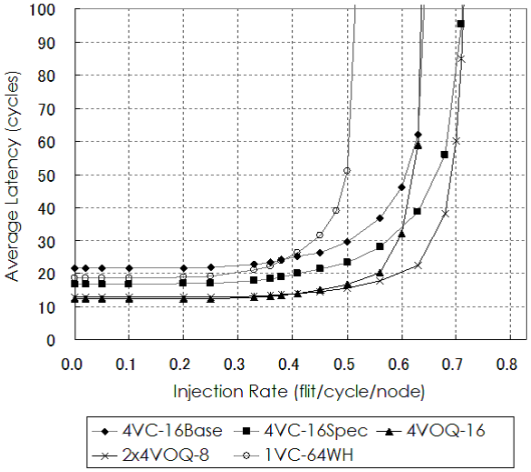

Fig. 10 Network latencies of routers with 16-flit buffer (4VC-16Base = four 16-flit VCs/port Baseline; 4VC-16Spec = four 16-flit VCs/port LA+Spec; 4VOQ-16 = four 16-flit VOQs/port; 2x4VOQ-8 = eight 8-flit VOQs/port; 1VC-64WH = one 64-flit VC/port).

5. Conclusion

In this paper, we have presented a low-latency router architecture based on multiple VOQ that reduces the traffic congestion at the output ports under a heavy network load and improves the throughput of the router significantly. In the manner of speculatively performing switch allocation and switch traversal in parallel, the router can perform a packet transfer in only one clock cycle. The experimental result showed that, the design brings not only the hardware simplicity, but also a significant increase of communication performance. The proposed architecture of multiple VOQ offers an alternative that allows to reduce the hardware amount by 15.6% and communication latency by 25% in comparison with the look-ahead speculative VC router. This fact indicates the ability of our architecture to provide a hardware system for practical implementations of low cost low latency NoC.

References

1) Benini, L. and De Micheli, G.: Networks on Chips: A New SoC Paradigm, *IEEE Computer*, Vol.35, pp.70–78 (2002).
2) Dally, W.J. and Towles, B.: Route Packets, Not Wires: On-Chip Interconnection Networks, *The 38th ACM Design Automation Conference*, pp.684–689, Las Vegas, Nevada, USA (2001).
3) Bjerregaard, T. and Mahadevan, S.: A Survey of Research and Practices of Network-on-Chip, *ACM Computing Surveys*, Vol.38 (2006).
4) Peh, Li-S. and Dally, W.J.: A Delay Model and Speculative Architecture for Pipelined Routers, *The Seventh International Symposium on High-Performance Computer Architecture*, pp.255–266 (2001).
5) Dally, W.J. and Towles, B.: *Principles and Practices of Interconnection Networks*, Morgan Kaufmann (2004).
6) Kim, J., Nicopoulos, C., Park, D., Narayanan, V., Yousif, M.S. and Das, C.R.: A Gracefully Degrading and Energy-Efficient Modular Router Architecture for On-Chip Networks, *Proc. International Symposium on Computer Architecture* (*ISCA'06*), pp.4–15 (2006).
7) Mullins, R., West, A. and Moore, S.: Low-Latency Virtual-Channel Routers for On-Chip Networks, *Proc. 31st Annual International Symposium on Computer Architecture*, pp.188–197 (2004).
8) Mullins, R., West, A. and Moore, S.: The Design and Implementation of a Low Latency On-Chip Network, *Proc. Asia and South Pacific Design Automation Con-*

ference (*ASP-DAC'06*), pp.164–169 (2006).
9) Bertozzi, D. and Benini, L.: Xpipes: A Network-on-Chip architecture for gigascale Systems-on-Chip, *IEEE Circuits and Systems Magazine*, pp.18–31 (2004).
10) Kumar, A., Peh, Li-S., Kundu, P. and Jha, N.K.: Express Virtual Channels: Towards the Ideal Interconnection Fabric, *Proc. International Symposium on Computer Architecture* (*ISCA'07*), pp.150–161 (2007).
11) Park, D., Das, R., Nicopoulos, C., Kim, J., Vijaykrishnan, N., Iyer, R. and Das, C.R.: Design of a Dynamic Priority-Based Fast Path Architecture for On-Chip Interconnects, *Proc. IEEE Symposium on High-Performance Interconnects* (*HOTI'07*), pp.15–20 (2007).
12) Michelogiannakis, G., Pnevmatikatos, D.N. and Katevenis, M.: Approaching Ideal NoC Latency with Pre-Configured Routes, *Proc. International Symposium on Networks-on-Chip* (*NOCS'07*), pp.153–162 (2007).
13) Matsutani, H., Koibuchi, M., Amano, H. and Yoshinaga, T.: Prediction Router: Yet Another Low Latency On-Chip Router Architecture, *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture* (*HPCA'09*), pp.367–378 (2009).
14) Kim, J., Park, D., Theocharides, T., Vijaykrishnan, N. and Das C.R.: A Low Latency Router Supporting Adaptivity for On-chip Interconnects, *Proc. 42nd annual Design Automation Conference* (*DAC'05*), pp.559–564 (2005).
15) Tamir, Y. and Frazier, G.: High performance multi-queue buffers for VLSI communication switches, *Proc. 15th Annual Symposium on Computer Architecture*, pp.343–354 (1988).
16) McKeown, N., Anantharam, V. and Walrand, J.: Achieving 100% Throughput in an Input-Queued Switch, *IEEE Trans. Comm.*, Vol.47, No.8, pp.1260–1267 (1999).
17) Nguyen, S.T. and Oyanagi, S.: A Low Cost Single-Cycle Router Based on Virtual Output Queuing for On-Chip Networks, *Proc. 13th Euromicro Conference on Digital System Design*, pp.60–67 (2010).

(Received May 6, 2010)
(Accepted August 28, 2010)

Son Truong Nguyen received his B.Sc. degree in Electrical Engineering from Le Quy Don Technical University, Vietnam in 1996 and his M.E. degree in Electrical and Computer Engineering from National Defense Academy, Japan in 2004. He is currently a doctoral student at the Graduate School of Science and Engineering, Ritsumeikan University, Japan. His research interests focus on networks-on-chip, interconnection networks and computer architectures. He is a member of IPSJ.

Shigeru Oyanagi is a professor at the Department of Computer Science, College of Information Science and Engineering, Ritsumeikan University, Japan. He received his M.E. and Ph.D. degrees from Kyoto University in 1974 and 1979, respectively. His research interests include parallel processing, computer architecture, database and data mining. He is a member of IEICE, IPSJ, ACM and IEEE.