Channel Buffer Organization of NoC Using Virtual Channel

Prathiba.K.N , Vijaya Vittala

Department of Electronics and Communication Engineering

Institute of Technology

Bangalore-560077

Abstract-The processing cores on the chip utilizes an efficient communication structure similar to NoC. The virtual channels in the NoC helps to recover the data flow and helps to improve the performance in the NoC system. Here DAMQs are used to achieve virtual channel control along with the buffer. The buffers slots are used according to the traffic in virtual channel. The efficient dynamic virtual channel (EDVC) has a new input port micro architecture in DAMQ buffers which also helps in link-list tables in the buffer organization. The EDVC input port organization helps in consuming of more amount of large power consumption which also improves NoC latency by high percentage gain and also the throughput. In this implementation the virtual channels includes different number of buffer slots which depends on the number of incoming data by leaving the performance merits of DAMQs has some limitations where, the input port micro architecture supports for efficient dynamic virtual channel.

IndexTerms-Adaptive virtual channel (VC), channel bufferorganization, on-chip communication, router microarchitecture, VC structure.

I. INTRODUCTION

Network on chip (NoC) architecture provides a communication infrastructure for the cores of a multicore system-on-chip. The communication mechanism involved in the NoCs has packet based wormhole routing. The messages in the wormhole routing is defined as the multiple packets where as a single packets has different number of flits, the header flit says the messages regarding the packet destination through the routers. Each port in the VCs, flits are temporarily kept in the input or output buffers which utilizes this for efficient dataflow and to share a physical communication channel through multiple packets in NoC systems. In this paper, we assume that a router implements VCs at the input ports as shown in Fig. 1 and the term buffer is used. In the wormhole routing the messages are organized as the multiple packets where each number of packets consist of varying number of flits/data.

Manuscript received June 11, 2014; revised October 5, 2014 and December 23, 2014; accepted February 13, 2015. Date of publication March 6, 2015; date of current version January 19, 2016. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and in part by the CAD tools through the CMC Microsystems.

The authors are with the Department of Electrical and Computer Engineering, Ryerson University, Toronto, ON M5B 2K3, Canada (e-mail: masoud.oveisgharan@ryerson.ca; gnkhan@ee.ryerson.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2015.2405933

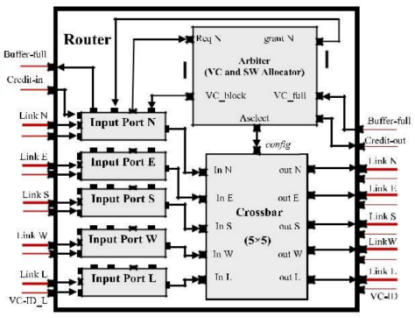

Fig. 1. 5 × 5 router architecture.

The above architecture shows the 5 x 5 router implementation in the NoC system. The FIFO and the queue are interchangeable which refers for all the types of buffers with first come first serve process which also involves FIFO buffers. There are static and dynamic virtual channels which has incoming packets (DAMQs), the BSs are dynamically allocated to the incoming packet flits. In this paper we assume that the router implements the virtual cannels as the input ports which is showed in the above figure. Each input port in the virtual channel helps for storage and temporarily stores the incoming data. In the FIFO the read and write pointers shows the location that here the data has to be read or written according to the pointers location. In the static input port the incoming and the outgoing of the data is not very much complex and takes a single step arbitration, whereas the incoming and outgoing of the data takes more than one step which may be lesser than three steps.

A. Dynamically Allocated Multiqueues

According to the survey in the virtual channels it is very difficult to manage the communication load. At this moment some of the virtual channels remains ideal and hence other virtual channels becomes heavier therefore the

allocation of the buffers should be in more numbers in order to avoid the ideal virtual channels.

DAMQ has the capacity to control the virtual channel with the maximum buffer and also helps for good results, along with this it also helps the FIFO to avoid the traffic and queue spacing across the output ports.

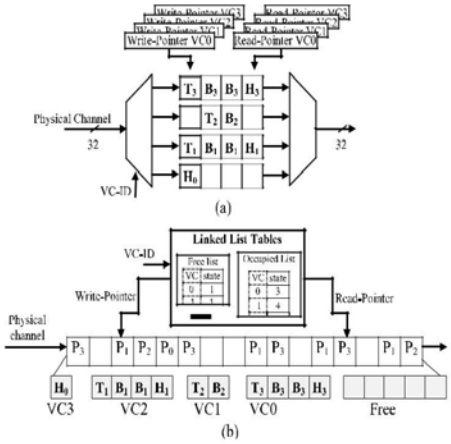

Fig. 2. Input ports with (a) static and (b) dynamic queues.

Depth keeps more flits of a packet and leads to a free route of the packet that will reduce contention. DAMQs have been used for VC organization in NoC systems. It can also resolve various NoC-related issues, such as contention and deadlock. However, DAMQs have a number of limitations listed below.

The first problem is the complex hardware due to link-list and dynamic queue management. The second problem is related to the queue structure that is tailored for deterministic routing.

It cannot look after fully adaptive routing since the routing decision for a new packet is made in conjunction with the output queues. With such flow control mechanism, the routing adaptively cannot be established. There are interventions among the VCs that can lead to higher traffic congestion as compared with static VCs.

The initial problems can be resolved by using DAMQ based virtual channels and the final stage problems are listed in this paper. The huge incoming and outgoing data will be having delays has mentioned above, this can be solved by using link-list and virtual channel regulator (ViCHAR) which has central table registers and helps for direct data flow. In the clock edge the registers get updated.

B. Static and Dynamic Buffer-Based Input Ports

The static and dynamic VC based input port micro architecture are shown in the figure 2. The each virtual channel in the static input port is built using a parallel FIFO buffer which is very simple the number of VCs is equal to the number of FIFOs used. The read and write pointers points the location of the FIFO where the data has to be written or read. The incoming and the outgoing of the data is very simpler in the static input port.

In the case of dynamic VC input port the VC buffers dynamically allocated using the link list based DAMQ seeing the traffic in the input ports. A single buffer can have the number of multiple virtual channels and the data is directed by link-list as shown in the figure 2(b).

If the arbitration takes in one step the incoming and the outgoing data takes five steps to execute.

In a link-list DAMQ-based VC, the read and write pointers cannot be updated at read or write events; instead, they can be updated one clock event after the read and write. Where as in the static virtual channels the pointers are incremented for every read and write events. It has the pipeline stages dynamically allocated multiqueues have one clock cycles at each input ports longer than the static virtual channel input ports.

II. PREVIOUS RESEARCH WORK

A link-list-based mechanism has been frequently utilized to implement DAMQ buffers. An DAMQWR uses DAMQ with some recruit registers to implement adaptive routing for on-chip communication. DAMQWR method has additional delays due to recruit register updates and operations.

B. Evripidou introduced a link-list-based DAMQ architecture with congestion awareness. They added congestion-avoidance logic to the arbiter for predicting congestion at immediate neighbors. Mingche et al. [2] also proposed DAMQ architecture to remove head of line (HoL) blocking. Their method to remove congestion is expensive compared with our EDVC approach. Zhang et al. [10] presented a novel multi-VC dynamically shared buffer for NoC systems. Their design employed a small prefetch buffer for each VC. This buffer can store the data read from the shared buffer to provide dedicated storage for each output port. They also proposed a fair credit management method to avoid the situation where a single VC can occupy the shared buffer exclusively. Liu and Delgado-Frias [11] proposed a new DAMQ buffer organization scheme with a reserved space for each of the VCs.

C. Another drawback of this approach is the high cost of its input port for some configurations. Several features of ViCHaR have encouraged some other researchers to employ it in their designs. Nicopoulos et al. [12] presented the design of an intelligent buffer that logically reorders the entries in an FIFO buffer to minimize the leakage power. In this design, the BSs are first classified on the basis of their leakage characteristics. Then, the write

module attempts to direct incoming flits to the least leaky slot. Xu et al. [4] employed ViCHaR, where VCs are assigned based on the designated output port of a packet in an effort to reduce HoL blocking. Their buffer design is similar to ViCHaR except that each VC can store multiple packets and the number of VCs is fixed. It uses a smaller arbiter, and their VC allocation scheme is hybrid.

The SBVC investigation was mainly conducted to investigate the effect of varying the number of VCs of a channel and their buffer depth on the overall NoC throughput. On the other hand, the EDVC approach being proposed in this paper demonstrates higher performance under high-contention scenarios as it handles the blockage effectively and for a much lower cost. The flit-data pointed by the read-pointer appear at the SRAM output. When the credit-in signal is activated, the flit-data are stored in the SRAM slot pointed by the write-pointer. The output of the read-pointer circuit (RD) points to the initial location of slot-state table that is set and remains there for one clock cycle

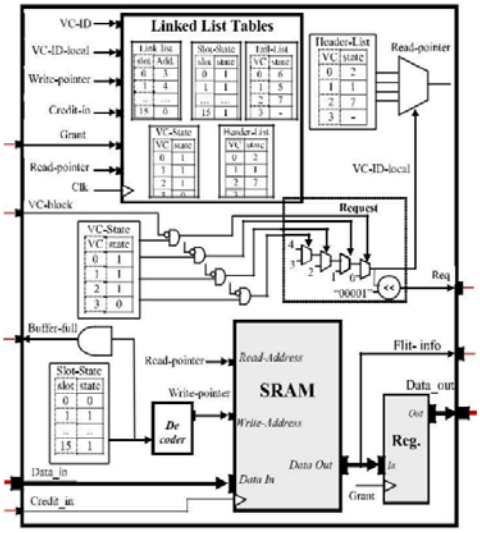

Fig. 3. Link-List-based CDVC input port.

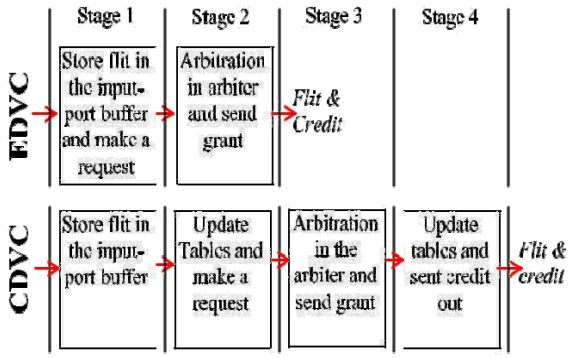

Fig.4. EDVC versus CDVC input-port pipelines

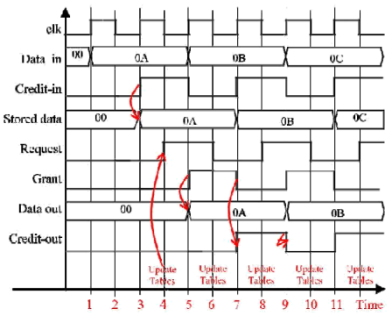

Fig. 5 Pipelined communication mechanism.

III. CONVENTIONAL DAMQ MICRO ARCHITECTURE

Types of schemes are used to design DAMQ mechanism, it consists link-list, self-compacting and ViCHAR. This method usually called conventional dynamic VC (CDVC).

By comparing the EDVC structure with CDVC we can illustrate the complexity and cost of this mechanism.

A. CDVC Router

The CDVC router has an input-port modules, an arbiter, and a crossbar switch. It consists of static random access memory (SRAM) buffer, five lookup tables, other logic circuits and ports. The flit size and the SRAM buffers slot size is equal and the read write pointer highlighted to point out the data at SRAM output.

B. CDVC Communication

The communication between the source, intermediate and destination routers is processed by credit signals-based handshaking. The source core ensures to send the flit of packet to generate a credit signal. In the location of destination router the information or data stored in input port buffer. Once the buffer is full it will acknowledge the source router to stop sending the data at the input port. The two clock events used to store the flit data and to transfer the data using pipelined communication. The negative clock edge is used to update the tables and the positive clock edge is used to detect the signal.

The DAMQ has a common features of DAMQ input port to create a dynamic flow control for the utilization of EDVC mechanism. For example, VC-ID is saved with the flit-data in the input-port buffer to assist EDVC mechanism in issuing the request signals to the arbiter. The two types of signals are VC-full and VC-block are used to maintain the order of flits associated with each VC. The incoming or outgoing flit works like a parallel FIFO. Assume there is no blockage, each incoming flit and its VC-ID is stored in the buffer location pointed by the write-pointer. The flit pointed by the read-pointer is read, arbitrated based on its VC-ID and sent out of the buffer. As mentioned earlier, the flit arrival/departure time in EDVC is the same as that of a static VC organization and

requires three steps. We employ asynchronous communication in our EDVC mechanism for NoCs. The following functions describe the working of EDVC in detail.

1) Flit Arrival (Clk-Edge #2): A credit-in signal causes theincoming flit and its VC-ID to be saved in a slot pointed by the write-pointer. Meanwhile, the corresponding bit of the slot-state table is set.

2) Request Signal (Clk-Edge #2): When the read-pointerpoints to a slot and its slot-state bit is set, a request signal is issued according to the VC-ID. The arbiter will read the flit information and perform arbitration.

3) Grant Signal (Clk-Edge #3): If the requested output portis open, the arbiter allocates the proper address for the crossbar switch and VC-ID before issuing a grant signal.

The novelty of our EDVC approach can be summarized as follows.

1) Our novel EDVC approach improves NoC performance considerably by adding a little hardware.

2) The EDVC mechanism employs logic circuits instead of tables to manage shared VC slots and to determine the next free write-slot and available read-slot. It saves one clock cycle for each flit arrival/departure from the input port.

3) The EDVC approach is much simpler as compared with table-based mechanisms. The main component of EDVC is the pointer circuits, which has a scalable structure to optimize its performance.

4) There are no configuration constraints in EDVC approach as compared with other DAMQ mechanisms. For example, the link-list approaches require a reserved space for each VC, and the ViCHaR buffer grows with more number of flits per packet.

5) The EDVC approach employs a simple congestion avoidance mechanism.

IV. EXPERIMANTAL RESULTS

Table 1: Hardware Specification of CDVC, EDVC and ViCHAR input-port

Type of input-port	ASIC design (90 nm)			FPGA design (Altera Stratix III)		
	Cell Area (μm^2)	Power (μW	Critical path delay ($nsec$)	Comb. Logic elements	Registers ($bits$)	F_{max} (MHz)
CDVC 4-slots	5809	218	1.57	95	112(64^b)	352
ViCHaR 4-slots	6646	319	1.56	75	132(64^b)	300
EDVC Fast-Read/Write 4-slots	4991	106 (60^a)	1.11	83	107(64^b)	384
EDVC Fast-Write 4-slots	4674	108	0.57	78	97(64^b)	1082
CDVC 8-slots	10328	370	1.54	180	204(128^b)	286
ViCHaR 8-slots	21274	1236	2.26	306	392 (128^b)	197
EDVC Fast-Read/Write 8-slots	9524	150 (88^a)	1.62	206	186(128^b)	244
EDVC Fast-Write 8-slots	8687	162	0.47	174	174(128^b)	851
CDVC 16-slots	19813	688	2.02	332	388(256^b)	271
ViCHaR 16-slots	48463	2968	2.76	548	896 (256^b)	175
EDVC Fast-Read/Write 16-slots	19448	240 (147^a)	2.22	441	340(256^b)	171
EDVC Fast-Write 16-slots	17016	263	0.92	324	327(256^b)	726
CDVC 32-slots	39174	1306	3.23	670	764(512^b)	218
ViCHaR 32-slot	109849	6989	2.86	1183	2040(512^b)	125
EDVC Fast-Read/Write 32-slots	40716	413 (262^a)	4.64	964	646(512^b)	118
EDVC Fast-Write 32-slots	34295	457	0.94	727	632(512^b)	597
Arbiter 4-VC	28380	1904	4.17	1116	240	127
Cross-bar	2502	611	-	160	-	-

A. Hardware Requirements and Parameters of Input Ports

The hardware requirements and characteristics of EDVC fast-write, fast-read/write, CDVC and ViCHaR input-port architectures are determined by employing Synopsys Design Compiler for generic 90-nm technology and Mentor Graphics ModelSim for Stratix-III FPGA. We have coded the microarchitectures using Verilog and simulation is performed by employing ModelSim to measure various performance metrics. All the input ports use a dual-ported SRAM buffering and we apply the same setup constrains. The Synopsys Design Compiler is used to measure the power consumption and IC area. The setup employs CMOS technology parameters of Synopsys 90-nm library, global operating voltage of 1.2 V, and time period of 5 ns (200 MHz). The width of slot buffer is equal to the flit size of 16 bits.

The characteristics of input port, arbiter, and crossbar are listed based on their buffer sizes in Table I. The EDVC

fast-write input-port buffer has the optimum area and timing characteristics. On average, the EDVC fast-write consumes 15% less IC area, 58% less power consumption, 64% less critical path delay, 4% less combinational logic elements, and 15% less registers as compared with CDVC.

The table also provides the amount of SRAM-based registers in the brackets. EDVC fast-write uses a simple read-pointer versus fast read-pointer used in EDVC fast-read/write approach. The simple read-pointer leads to 9% higher power consumption of fast-write versus fast-read/write. In spite of optimal hardware, EDVC fast-write has lower performance than EDVC fast-read/write that is also investigated. The CDVC and ViCHaR architectures are table-based designs, where a register-based control table direct the operation of mechanisms [6]–[8]. Registers and/or latches are updated on one edge (or level) of clock causing the table-based mechanisms to take one clock longer than EDVC approach. ViCHaR is expensive in terms of hardware due to large control table. Equations (1) and (2) provide the Size of Control Table (SCT) for CDVC and ViCHaR techniques.

$$SCTCDVC = SB.\ln(SB) + SB + VC + 2.VC.\ln(SB) \quad (1)$$

$$SCTViCHaR = SB.\ln(SB).FP + SB + 2.SB.\ln(FP) \quad (2)$$

Where,

SB number of slots per port buffer; FP number of flits per packet; VC number of VCs.

B.EDVC Performance Evaluation

Application specific and hotspot traffic patterns:

In the first experiment we simulated and tested EDVC mechanism for two different traffic patterns, including application-specific and hotspot traffic patterns.

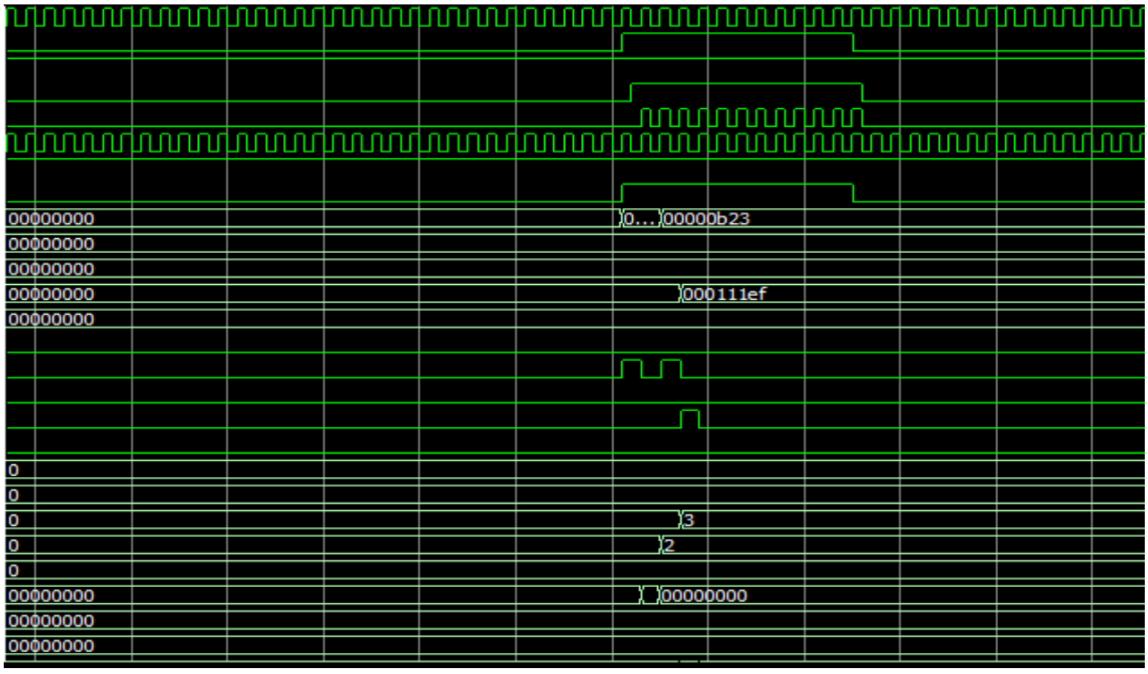

Fig. 6.Top clock gated simulation

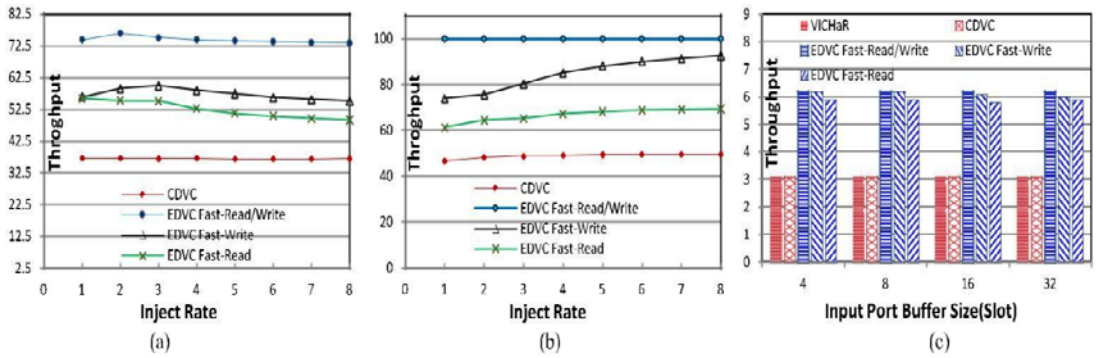

Fig. 7. Average throughput. (a) MPEG4 decoder. (b) AV benchmark. (c) Hotspot traffic.

For application-specific traffic, two NoC applications:

1) MPEG4 decoder and 2) audio–video (AV) benchmarks are tested for 3 × 4 and 4 × 4 mesh topology NoCs respectively. For hotspot traffic, one destination is chosen for all the source cores during a time period. By evaluating the results of these three traffic patterns, we demonstrate the efficiency of our EDVC approach in terms of throughput and latency. The NoC topology selected is mesh and packet communication follows the X Y routing. The communication of packets is based on wormhole switching where the channel width is equal to the flit size (16 bits). Each packet is made of 16 flits. Each input port utilizes 2 VCs and 8-slots per input-port buffer in the application-specific traffic. In the case of hotspot traffic, each input port has 4 VCs and the buffer depth is varied from 4 to 32 slots. The link delay between two routers is negligible as compared with the delay of a router and it is ignored. Source, destination cores, and routers operate at the same clock rate. For hotspot traffic pattern, all the source cores send their packets to one destination (e.g., destination core#10).

We measured throughput and latency where throughput is measured by the rate of packets received to the maximum number of packets being injected at a specific time. The average latency is measured by the average time delays (per clock cycle) associated with the departure and arrival of a Specific number of packets in the NoC. We apply different packet injection rate to measure the performance in the application-specific traffic. Packet injection rate is changed per time unit. The time unit is determined based on the maximum bandwidth of the source cores.EDVC fast-read/write has lower latency than that of CDVC for all the injection rates. However, it is higher for the EDVC fast-read or fast-write. For MPEG4, seven source cores send data to one destination core#5 causing high contention. High contention results in higher latency for EDVC fast-read and fast-write as compared with CDVC. For AV benchmark, the NoC communication is less congested as a maximum of four cores send packet to destination core#10.

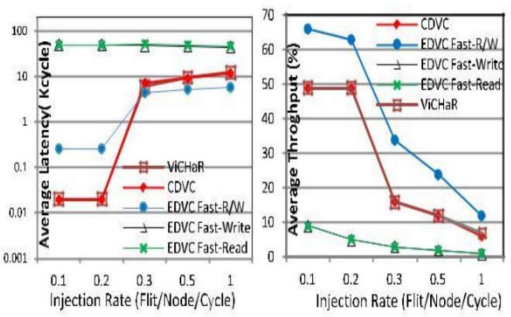

Fig. 8. Average latency and throughput for Complement traffic

In the case of hotspot traffic, the contention is high. Only one destination receives all the data, and all the flits have to wait around that destination core to be served. A slower flit movement speed in the NoC under such condition does not improve performance of EDVC fast-read and fast-write mechanisms. However, the two times faster clock cycle of fast-read and fast-write mechanism causes a bit higher performance.

V. CONCLUSION

The microarchitecture of conventional DAMQ input port for NoC routers has high NoC latency. We have introduced three novel DAMQ-based input-port microarchitectures (EDVC) requiring lower hardware resources. The EDVC approach is thoroughly investigated and the evaluation results are presented showing its efficiency in terms of performance and hardware. On average, an EDVC input port consumes less registers for FPGA implementation and less power for ASIC design. It also has higher Fmax as compared with the CDVC input port for small buffer sizes. Moreover, the EDVC mechanism indicated better latency and throughput by and, respectively, as compared with CDVC approach for application-specific traffic in the NoC system. It also shows better performance for hotspot and complement traffic patterns for 4× 4 and 8 × 8 mesh topologies as compared with CDVC and ViCHaR techniques.

REFERENCES

[1] W. J. Dally and B. Towles, “Bufferd flow control,” in Principlesand Practices of Interconnection Networks. San Francisco, CA, USA:Morgan Kaufmann, 2003.

[2] L. Mingche, G. Lei, S. Wei, and W. Zhiying, “Escaping from block-ing: A dynamic virtual channel for pipelined routers,” in Proc. Int.Conf. Complex, Intell., Softw. Intensive Syst., Barcelona, Spain, 2008,pp. 795–800.

[3] Y. Choi and T. M. Pinkston, “Evaluation of queue designs for true fully adaptive routers,” J. Parallel Distrib. Comput., vol. 64, no. 5,pp. 606–616, May 2004.

[4] Y. Xu, B. Zhao, Y. Zhang, and J. Yang, “Simple virtual channel allocation for high throughput and high frequency on-chip routers,” in Proc. IEEE 16th Int. Symp. High Perform. Comput. Archit., Bengaluru,India, Jan. 2010, pp. 1–11.

[5] M. Lai, Z. Wang, L. Gao, H. Lu, and K. Dai, “A dynamically-allocated virtual channel architecture with congestion awareness for on-chip routers,” in Proc. 45th ACM/IEEE DAC, Anaheim, CA, USA, Jun. 2008,pp. 630–633.

[6] M. Evripidou, C. Nicopoulos, V. Soteriou, and J. Kim, “Virtualizing virtual channels for increased

network-on-chip robustness and upgrade-ability," in Proc. IEEE Comput. Soc. Annu. Symp. VLSI, Amherst, MA, USA, Aug. 2012, pp. 21–26.

[7] G. L. Frazier and Y. Tamir, "The design and implementation of a multiqueue buffer for VLSI communication switches," in Proc. IEEEICCD, Cambridge, MA, USA, Oct. 1989, pp. 466–471.

[8] C. A. Nicopoulos, D. Park, J. Kim, N. Vijaykrishnan, M. S. Yousif, and C. R. Das, "ViChaR: A dynamic virtual channel regulator for network-on-chip routers," in Proc. 39th Annu. IEEE/ACM Int. Symp.Microarchitecture, Orlando, FL, USA, Dec. 2006, pp. 333–346.

[9] Y. Tamir and G. L. Frazier, "Dynamically-allocated multi-queue buffers for VLSI communication switches," IEEE Trans. Comput., vol. 41, no. 6,pp. 725–737, Jun. 1992.

[10] H. Zhang, K. Wang, Y. Dai, and L. Liu, "A multi-VC dynamically shared buffer with prefetch for network on chip," in Proc. IEEE 7th Int. Conf.Netw., Archit., Storage, Xiamen, China, Jun. 2012, pp. 320–327.

[11] J. Liu and J. G. Delgado-Frias, "DAMQ self-compacting buffer schemes for systems with network-on-chip," in Proc. IEEE ICCD, Las Vegas, NV, USA, 2005, pp. 97–103.

[12] C. Nicopoulos, A. Yanamandra, S. Srinivasan, N. Vijaykrishnan, and M. J. Irwin, "Variation-aware low-power buffer design," in Proc. Conf.Rec. 41st Asilomar Conf. Signals, Syst., Comput., Pacific Grove, CA,USA, 2007, pp. 1402–1406.

[13] M. O. Gharan and G. N. Khan, "A novel virtual channel implementation technique for multi-core on-chip communication," in Proc. WAMCA, New York, NY, USA, 2012, pp. 36–41.

[14] J. Liu and J. G. Delgado-Frias, "A shared self-compacting buffer for network-on-chip systems," in Proc. 49th IEEE Int. Midwest Symp.Circuits Syst., San Juan, Puerto Rico, Aug. 2006, pp. 26–30.

[15] J. Park, B. W. O'Krafka, S. Vassiliadis, and J. Delgado-Frias, "Design and evaluation of a DAMQ multiprocessor network with self-compacting buffers," in Proc. Supercomputing, Washington, DC, USA, 1994, pp. 713–722.

[16] M. O. Gharan and G. N. Khan, "Flexible simulation and modeling for 2D topology NoC system design," in Proc. 24th CCECE, Niagara Falls, ON, Canada, May 2011, pp. 000180–000185.

[17] N. Alfaraj, J. Zhang, Y. Xu, and H. J. Chao, "HOPE: Hotspot congestion control for Clos network on chip," in Proc. 5th IEEE/ACM NoCS, Pittsburgh, PA, USA, May 2011, pp. 17–24.

[18] V. Dumitriu and G. N. Khan, "Throughput-oriented NoC topology generation and analysis for high performance SoCs," IEEE Trans.Very Large Scale Integr. (VLSI) Syst., vol. 17, no. 10, pp. 1433–1446,Oct. 2009.

[19] R. S. Ramanujam, V. Soteriou, B. Lin, and L.-S. Peh, "Design of a high-throughput distributed shared-buffer NoC router," in Proc. 4thIEEE/ACM Int. Symp. NoCS, Grenoble, France, May 2010, pp. 69–78.