

# Realization of Efficient High Throughput Buffering Policies for Network on Chip Router

**Liyaqat Nazir**

National Institute of Technology, Srinagar, J&K, India.  
E-mail: [liyaqat\\_02phd13@nitsri.net](mailto:liyaqat_02phd13@nitsri.net)

**Rohie Naaz Mir**

National Institute of Technology, Srinagar, J&K, India  
E-mail: [naaz310@yahoo.co.in](mailto:naaz310@yahoo.co.in)

**Abstract**—The communication between processing elements is suffering challenges due to power, area and latency. Temporary flit storage during communication consumes the maximum power of the whole power consumption of the chip. The majority of current NoCs consume a high amount of power and area for router buffers only. Removing buffers and virtual channels (VCs) significantly simplifies router design and reduces the power dissipation by a considerable amount. The buffering scheme used in virtual channeling in a network-on-chip based router plays a significant role in determining the performance of the whole network-on-chip based mesh. Elastic buffer (EB) flow control is a simple control logic in the channels to use pipeline flip-flops (FFs) as storage locations. With the use of elastic buffers, input buffers are no longer required hence leading to a simplified router design. In this paper properties of buffers are studied with a test microarchitecture router for several packet injection rates given at an input port. The prime contribution of this article is the evaluation of various forms of the elastic buffers for throughput, FPGA resource utilization, average power consumed, and the maximum speed offered. The article also gives a comparison with some available buffering policies against throughput. The paper presents the synthesis and implementation on FPGA platforms. The work will help NoC designers in suitable simple router implementation for their FPGA design. The implementation targets Virtex5 FPGA and Stratix III device family.

**Index Terms**—Network-on-chip, virtual channels, buffers.

## I. INTRODUCTION

In the past few years, with the concept of Network-on-Chip communication architecture, NoC has attracted a lot of attention by providing higher bandwidth and higher performance architectures for communication on chip [23]. NoC can provide simple and scalable architectures if implemented on reconfigurable platforms [1]. Network on chip offers a new communication paradigm for system

on chip (SoC) design [2]. Many processing elements of SoC are connected through Network-on-chip (NoC) routers which are arranged in some regular fashion such as Mesh, linear, torous, 2D, 3D type of topologies. To achieve high performance, the router should provide high band width and low latency [3]. Although the performance of the NoC is normally seen by its throughput, which is defined by the network topology, router throughput and the traffic load at the network [4]. The routers for a NoC must be designed to meet latency and throughput requirements amidst tight area and power constraints; this is a primary challenge designer are facing as many-core systems scale. As router complexity increases with bandwidth demands, very simple routers (unpipelined, wormhole, no VCs, limited buffering) can be built when high throughput is not needed, so area and power overhead is low. Challenges arise when the latency and throughput demands on on-chip networks are raised [32]. A router's architecture determines its critical path delay which affects per-hop delay and overall network latency. Router microarchitecture also impacts network energy as it determines the circuit components in a router and their activity. The implementation of the routing, flow control and the actual router pipeline will affect the efficiency at which buffers and links are used and thus overall network throughput [23]. The area footprint of the router is clearly determined by the chosen router microarchitecture and underlying circuits. The critical path of the data path units in the router and the efficiency of control path units determine the router throughput [5-8]. The data paths of the on-chip router comprise of buffers, VC and switching fabric and the control paths of on-chip communication routers are largely composed of arbiters and allocators [9]. Allocators are used to allocate virtual channels (VC) and to perform matching between a groups of resources on each cycle [10-12]. Upon the flit arrival at the input port, contention for access to the fabric with cells at both input and output occurs. The router units exchange necessary handshake signals for data/flit transfer as shown in figure 1. A VC allocator thus performs allocation between the input flits and allows at most one flit contending at the input port to be destined to the selected output port [13]. In order to reduce the line of blocking, the rest of the contending flits are buffered into the virtual channels or

buffers of the router so as to service them in coming appropriate clock cycles [19]. Elimination of input buffers eliminates the need of virtual channels (VCs). This increases head-of-line blocking and causes reduction of performance. However, area and power are also reduced by eliminating use of buffers [20]. Hence to eliminate buffer cost, elastic buffer (EB) flow control has been proposed in [20-21]. Since the handshake signals allows the sender and the receiver to stop their operation for an arbitrary amount of time. Therefore buffering should be implemented in both sides to keep the available data that cannot be consumed during a stall in either side of the link. The elastic buffers are simple control logic based channels that can be attached at both sending router and receiving router. The elastic buffers are capable of implementing dual interface i.e. it enqueues new data by its internal logic from the incoming link and dequeues the available data to the outgoing link, when the valid and ready signals are appropriately asserted. Similarly, at receiver side elastic buffer enqueues when it is ready to receive a flit and drains the stored flit to its internal logic. Owing to its elastic operation, based on simple ready valid handshake signals, elastic buffers is a primitive and simplified form of NoC buffering, which can be easily integrated in a plug-and-play manner at the inputs and outputs of the routers (or inside them), as well as on the network links to act as a buffered repeater [22].

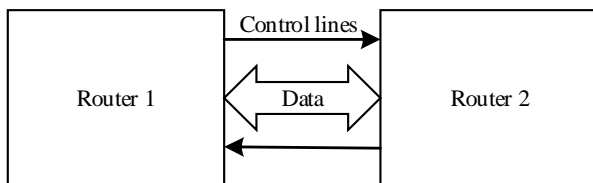


Fig.1. Block Diagram of NoC Router Communication.

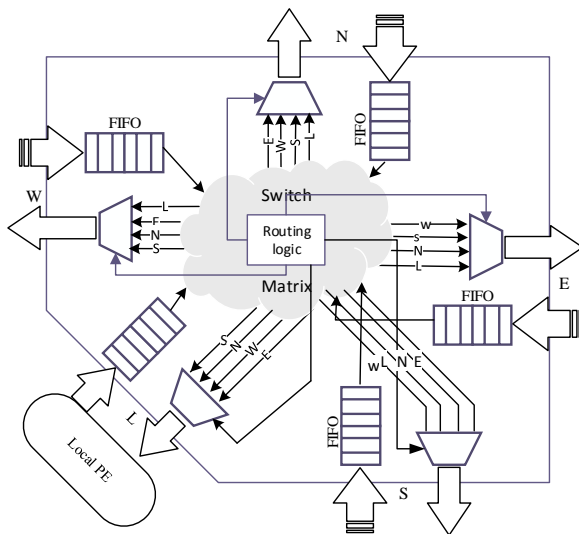


Fig.2. Block Diagram of NoC Router.

This work focuses on the implementation and evaluation of low vector density elastic buffers that can be used in Network on chip router implemented on reconfigurable platforms. The word length selected in this work is the 8-bit vector. In this paper, we carried out

implementation and synthesis of various elastic buffering policies used in a NoC microarchitecture, evaluated it for various parameters of interest like throughput, power dissipation, speed, Area delay product (ADP), power delay product (PDP) and resources utilized on Virtex 5 and Stratix III FPGA platforms and compared it with available work done. The rest of the paper is organized as follows: Section 1 gives the introduction, section 2 mentions briefly related work carried in this area, section 3 defines elastic channels and buffers briefly, section 4 talks about operational details, section 5 talks about synthesis and implementation, section 6 provides the conclusion and future scope of the work carried out and references are listed at the end.

## II. RELATED WORK

In NoC based architectures buffering policies play a key role in determining the throughput, latency, utilization and energy consumption. A considerable research effort has been devoted to buffering policies that can be adopted in a NoC router microarchitecture in the last few years. While these policies focus mainly on energy efficiency and latency but they also increase the complexity of the router. Throughput, the key parameter of the NoC router needs to be maintained while reducing the complexity of router design. Sophisticated input-buffered routers have been proposed for extending throughput. For instance, the work in [24] proposes a flit-reservation flow control, which sends control flits ahead of data flits, and timestamps these control flits so that buffers can be allocated just-in-time when data flits arrive. However, this still relies on input buffers. The improvement of the congestion of incoming packets can be also checked by the virtual channel (VC) scheme as presented in work [26, 27]. Virtual channel scheme multiplexes a physical channel using virtual channels (VCs), leading the reduction in latency and increase in network throughput. The insertion of VCs also enables to implement policies for allocating the physical channel bandwidth, which enables support of quality of service (QoS) in applications [26]. The low latency virtual router presented in [28] addresses the blocking problem of the flits at router input and latency of the packets is also considered. Although having an acceptable level of latency, the throughput is less. Moreover, there have been several input-buffered router proposals, which target mainly network latency, making single-cycle routers feasible, such as speculative allocation [29-31]. Recently proposed architectures such as the concentrated mesh (CMESH) [33], the express virtual channel (EVC) [35], Asynchronous Bypass Channels (ABC) [36], Distributed Shared-Buffer Routers (DBSBR) [37] and the flattened butterfly (FBFLY) [34] bypass intermediate routers in order to provide good performance and attempt to achieve an ideal latency i.e. the wire delay from the source to its destination. However, these architectures also add complexity to the design of the on-chip network.

### III. ELASTIC CHANNELS AND BUFFERS

The Analysis and study of throughput and delay of networks with finite buffers have also been facilitated by queueing theory. Throughput and delay problems can be seen to be similar since the packets can be viewed as customers and the delay due to packet loss in the link as the arbitrary service time. Also, the phenomenon of packet overflow in the network can be modeled by a blocking after service method in stochastic networks. However, there is a subtle difference in the packet-customer analogy when the network has nodes that can send packets over multiple paths to the destination. Under such scenario, the node can choose to duplicate packets on both the paths, an event that cannot be captured directly in the customer-server based queueing model. However, that is not the case in line networks. Therefore, the problem of finding buffer occupancy distribution and consequently throughput and delay in certain networks is then seen to be identical to determining certain arrival/departure processes in an open stochastic network of a given topology [38-41]. Because the flits do not flow freely in the network, but are constrained by the link access scheduling discipline. Therefore, the lossless networks in which flits are never dropped is an idealistic assumption. Practically each link must also implement back pressure flow control, ensuring that a flit can only be transmitted on a channel if the receiving end has free buffer space [42]. This introduces an extra layer of admission control to the trans-receiver logic or the communicating node. This type of packet flow in the field of queueing theory consider a continuous time model for arrival and departure of packets in the network [43].

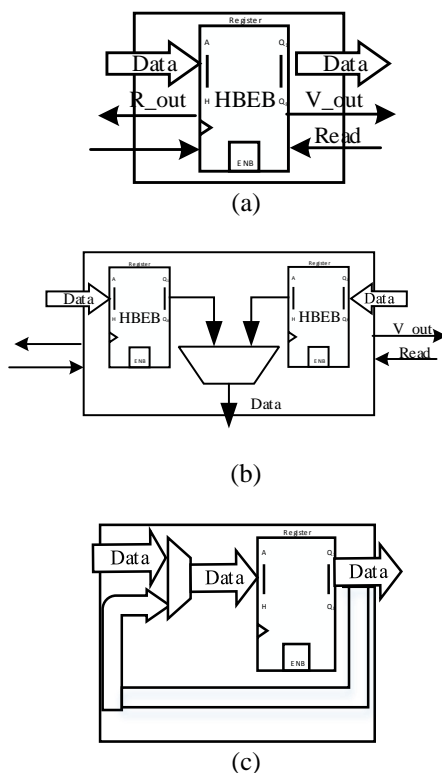


Fig.3. Block Diagram of (a)HBEB (b) FBEB (c)PEB (d)BEB

An abstract FIFO provides a push and a pop interface and informs its connecting modules when it is full or empty. A push (write) is done when valid data are present at the input of the FIFO and the FIFO is not full. At the read side, a pop (read) occurs when the upstream channel is ready to receive new data and the FIFO is not empty, i.e., it has valid data to send. The EBs implement the elastic protocol by replacing any simple data connection with an elastic channel. When an EB can accept an input, it asserts its ready signal upstream; when it has an available output, it asserts the valid signal downstream. When two adjacent EBs both see that the valid and ready signals are both true, they independently know the transfer has occurred, without negotiation or acknowledgment [20]. Block diagram of this protocol is shown in Fig 1. Figure 2 shows the simple architecture of a NoC router microarchitecture designed with the use of buffers at each communicating port.

### IV. OPERATIONAL DETAILS

Buffer organization has a large impact on network throughput, as it heavily influences how efficiently packets share link bandwidth. Essentially, buffers are used to house packets or flits when they cannot be forwarded right away onto output links. Flits can be buffered on the input ports and on the output ports. Output buffering occurs when the allocation rate of the switch is greater than the rate of the channel. We have considered four different buffering policies for throughput evaluation in our router design, the details of which are given in the subsections below. More emphasis has been on the rate that the message traffic can be sent across the router node so throughput is more appropriate metric. Throughput can be defined in a variety of different ways depending on the specifics of the implementation.

$$TP = \frac{(Total\ messages\ completed) \times (Message\ Length)}{(Number\ of\ Nodes) \times (Total\ time)} \tag{1}$$

Total messages completed refers to the number of whole messages that are successfully delivered. Message length is measured in flits. Total time is the time in clock cycles that elapses between the occurrence of the first message generation and the last message reception.

Throughput in our work is simply defined as the rate at which the packets are delivered/ accepted by the network for the traffic pattern fed to the input of the router.

#### A. Half bandwidth elastic buffer

The simplest form of an elastic buffer can be designed using one data register and letting the elastic buffer practically act as a 1-slot FIFO queue. Besides the data register for each design, we assume the existence of one state flip-flop F that denotes if the 1-slot FIFO is Full (valid\_out is asserted) or Empty (ready\_in is not asserted). The state flip-flop actually acts as an R-S flip flop. It is set (S) when the elastic buffer writes a new valid data (push) and it is reset (R) when data are popped out of the 1-slot FIFO. Figure 3(a) shows the block diagram of the

primitive elastic buffer. The Half-Bandwidth EB (HBEB) allows either a push or a pop to take place in each cycle, and never both. This characteristic imposes 50% throughput on the incoming and the outgoing links, since each EB should be first emptied in one cycle and then filled with new data in the next cycle. Thus, we call this EB a Half-Bandwidth EB (HBEB)[21]. The running data transfer table that passes through an HBEB is shown in table 1. The HBEB, although slower in terms of throughput, is very scalable in terms of timing. Every signal out of the HBEB is driven by a local register and no direct combinational path connects any of its inputs to any of its outputs, thus allowing the safe connection of many HBEB in series.

Table 1. Running Data Stream for 1-slot eb.

Cycle		0	1	2	3	4	5	6
Buffer write	Ready_out	1	0	1	0	0	1	0
	write	1	1	1	1	1	1	1
	Data	Data1	Data2	Data2	Data3	Data3	Data3	Data4
HBEB		Empty	Data1	Empty	Data2	Data2	Empty	Data3
Buffer Read	Read	1	1	1	0	1	1	1
	Valid_out	0	1	0	1	1	0	1
	Data	Empty	Data1	Empty	Data2	Data2	Empty	Data3

Table 2. Running Data Stream for 2-slot EB.

Cycle		0	1	2	3	4	5	6
Buffer write	Ready_out	1	1	1	1	0	1	1
	write	1	1	1	1	1	1	1
	Data	Data1	Data2	Data3	Data4	Data5	Data5	Data6
HBEB1		Empty	Data1	Empty	Data3	Data3	Empty	Data5
HBEB2		Empty	Data1	Data2	Empty	Data4	Data4	Empty
Buffer Read	Read	1	1	1	0	1	1	1
	Valid_out	0	1	0	1	1	0	1
	Data	Empty	Data1	Data2	Data3	Data3	Data4	Data5

Table 3. Running Data Stream for Pipeline eb.

Cycle		0	1	2	3	4	5	6
Buffer write	Ready_out	1	0	1	0	0	1	0
	write	1	1	1	1	1	1	1
	Data	Data1	Data2	Data3	Data4	Data5	Data5	Data6
HBEB		Empty	Data1	Data2	Data3	Data4	Data4	Data5
Buffer Read	Read	1	1	1	0	1	1	1
	Valid_out	0	1	0	1	1	0	1
	Data	Empty	Data1	Data2	Data3	Empty	Data 4	Data5

#### B. Full bandwidth elastic buffer

The half bandwidth elastic buffer has serious throughput limitations. The throughput limitation of HBEB can be resolved by adding two of them in each EB stage and using them in a time-multiplexed manner. Thus in each cycle, data are written in one HBEB and read out from the other HBEB alternatively thus providing means for the upstream and the downstream channel of 100% of write/read throughput. The design of the 2-slot EB that consists of two HBEBs needs some additional control

logic that indexes the read and writes position, the block diagram of the 2-slot EB is shown in table 2. When new data are pushed in the buffer they are written in the position indexed by the 1-bit EB pointer; on the same cycle, the EB pointer is pointed to the next available buffer. Equivalently, when new data is dequeued the head pointer is inverted. The 2-slot EB has valid data when at least one of the HBEB holds valid data and it is ready when at least one of the two HBEBs is ready. The incoming valid and ready signals are transferred via de-

multiplexers to the appropriate HBEB depending on the position shown by the head and tail pointers. This results in the 2-slot EB offering 100% throughput of operation, fully isolating the timing paths between the input and output handshake signals and constitutes a primitive form of buffering for NoCs [22]. A running data stream for the 2-slot EB connecting two channels is shown in the table 2.

### C. Pipelined and Bypass elastic buffer

2-slot EBs only are not necessary for a higher throughput this can be achieved even with 1-slot buffers that introduce a throughput-timing scalability tradeoff. The 1-slot EBs can be designed by extending the functionality of the HBEB in order to enable higher read/write concurrency. The first approach focuses on an increase in the concurrency on the write port (push) of the HBEB. New data can be written when the buffer is empty (as in the HBEB) or if it becomes empty in the same cycle. The PEB is ready to load new data even when at Full state, given that a pop (ready\_in) is requested in the same cycle, thus offering 100% of data-transfer throughput. The second approach called a bypass EB [25], offers more concurrency on the read port. In this case, a pop from the EB can occur even if the EB does not have valid data, assuming that an enqueue is performed in the same cycle. The bypass condition is only met when the EB is empty. Both buffers solve the low bandwidth problem of the HBEB and can propagate data forward at full throughput.

## V. SYNTHESIS AND IMPLEMENTATION

The evaluation the performance of Buffering policies on the target FPGA platforms has been carried through the Design ISE tool from Xilinx and Quartus design tool from Altera respectively. The RTL synthesis is carried out on Xilinx ISE using Xilinx Synthesis Technology (XST) tool. The XST tool reports the area results in terms of 'Number of Slice LUT's' and 'Number of Slice Registers'. We do not present the results in metrics such as logic cell count, gate count or ASIC area size, since the design is made for the reconfigurable platforms, and the conversions from FPGA metrics are not meaningful. The different buffering policies in this work have been evaluated for synthetic traffic patterns that are generated by the design tool.

### A. Methodology

The implementation of this work is targeted for Virtex 5 FPGA family from Xilinx and Stratix III FPGA device from altera. Only VLX from series has been considered as it is appropriate for general logic applications. The implementation is carried out for input data vector of 8-bits corresponding to the 4-bit handshake signals. The parameters considered are resource utilization, throughput, power delay product (PDP), area-delay product (ADP), timing and average power dissipation. Resource utilization is considered in terms of on chip FPGA components used. Timing refers to the clock speed

of the design and is limited by the setup time of the input and output registers, clock to the output time associated with the flip-flops, propagation and routing delay associated with critical paths, skew between the launch and capture register. Timing analysis is carried out by providing appropriate timing constraints. The average power dissipation mainly composes of dynamic power dissipation besides negligible static power dissipation. In order to provide a fair comparison of the results obtained the test bench is provided with same clock period and same input statistics in all the topologies implemented. The constraints related to the period and offset is duly provided in order to ensure complete timing closure. The design synthesis, implementation, and mapping have been carried out in Xilinx ise 12.4 [18]. Power metrics are measured using Xpower analyzer. The simulator data base is observed to obtain clock period and operating frequency of the implemented design.

### B. Experimental results

Resource utilization is actually FPGA resource utilization and Table 4 gives the comparison of the resource utilization of various elastic buffering policies implemented on Virtex5 FPGA device. The various resources utilized and the total resource utilization for the Virtex 5 device are plotted in figure 5. It can be seen that BPEB on account of high throughput utilizes the maximum on chip resources while as the HBEB being simple logic structure with throughput constraints utilizes the least on chip resources. Table 5 gives the comparison of the resource utilization of various Elastic buffering policies implemented on Stratix III FPGA device. Table 6 provides a comparison of maximum achievable clock rates post implementation for an 8-bit data elastic buffer. From Table 6, we can see that the HBEB can operate at maximum frequency of 1027.908 MHz, while as FBEB, PEB and BEB operates at 890.27, 713.47, 874.891 MHz respectively on Virtex5 FPGA, the corresponding operating frequencies on Stratix III device were measured as 1027.30 MHz, 1003.03 MHz, 760.47 MHz and 854.81 MHz for HBEB, FBEB, PEB, and BEB respectively. Although PEB having higher throughput, proves to be slowest in terms of maximum operating frequency in the elastic buffering policy. Finally static and dynamic power dissipation for the implemented buffering policy is considered. The static power dissipation in an FPGA consists of device static power dissipation and the design static power dissipation. Design static power although is very small percentage of dynamic power dissipation. The dynamic power dissipation is function of input voltage  $V^2$ , the clock freq ( $f_{clk}$ ), the switching activity ( $\alpha$ ), load capacitance ( $C_L$ ) and number of elements used. The analysis was done on a constant supply voltage and at maximum operating frequency for each structure. To ensure a reasonable comparison, diverse input test vectors in post route simulation were selected to represent worst case switching activity. The physical constraint file and design node activity file captured during the post route simulation were used for power analysis in Xpower analyzer tool. Table 7 shows the power dissipated in 1-

slot and 2-slot elastic buffer. The power dissipated in the clocking resources varies with the clocking activity (clock frequency) as provided in the PCF. The power-delay product measured for these buffering policies are plotted in figure 4. Similarly, the area-delay product (ADP) is plotted in figure 7. It can be seen from figure 7 that the HBEB measures least area-delay product (ADP) and BPEB has maximum ADP. While as from figure 8 we can see that FBEB measures least power delay

product (PDP) and PDP parameter for the HBEB is maximum. Figure 9 shows the throughput saturation at percent injection load for the buffering policies used in this paper and compares it with other existing buffering policies. The throughput saturation at percent injection load is relatively higher for the considered buffering policies as compared to the others mentioned against the comparison.

Table 4. Resource Utilization Comparison for HBEB and FBEB on Virtex 5 FPGA.

On chip Resource	Elastic buffer			
	Half bandwidth	Full bandwidth	Pipelined EB	Bypass EB
No of Slice registers	1	20	17	26
No of Slice LUTs	1	24	8	18
No. of occupied Slices	2	8	5	14

Table 5. Resource Utilization Comparison for HBEB and FBEB on STRATIX III FPGA.

On chip Resource	Elastic buffer			
	Half bandwidth	Full bandwidth	Pipelined EB	Bypass EB
Combinational ALUTs	9	16	17	21
Dedicated registers	9	20	22	23
Total registers	9	20	21	23
No. of occupied slices	3	N.A	5	6

Table 6. Timing Comparison for HBEB and FBEB on Virtex 5 Device

Timing Parameter	Elastic buffer			
	Half bandwidth	Full bandwidth	Pipelined EB	Bypass EB
Maximum frequency (MHz).	1027.908	890.27	713.47	874.891
Min available offset-in (ns).	1.442	1.683	1.784	4.592
Min available offset-out (ns).	3.337	3.55	3.361	8.096
Minimum period (ns)	1.362	1.123	1.402	1.143

Table 7. Power Dissipation for HBEB and FBEB on Virtex 5 Device.

FPGA Resource	Power dissipation (mW)			
	Encoding topology			
	Half bandwidth	Full bandwidth	Pipelined EB	Bypass EB
Clock	3.45	4.13	12.94	17.05
logic	0.05	0.14	.010	0.08
Signals	0.23	0.81	0.26	0.24
I/Os	15.85	9.73	4.48	1.94
Dynamic	19.58	14.81	17.78	19.31
Quiescent	3459.54	3459.02	3459.17	3459.25
<b>Total</b>	<b>3478.84</b>	<b>3473.83</b>	<b>3476.95</b>	<b>3478.46</b>

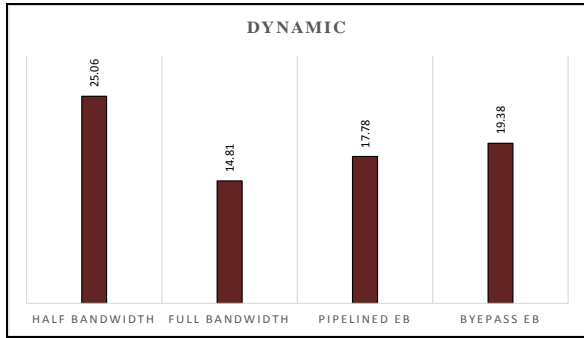


Fig.4. Plot of Dynamic Power used in Various Elastic Buffers on Virtex5 Device.

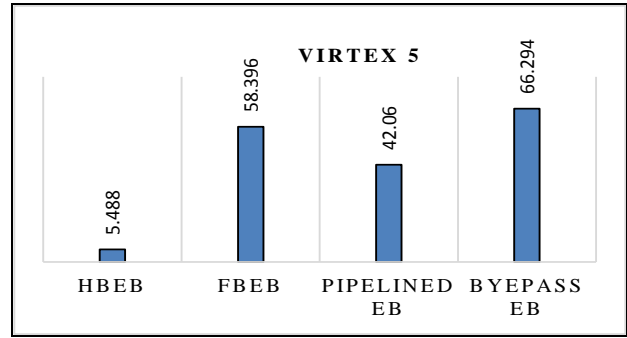


Fig.7. Plot of ADP in Various Elastic Buffers on Virtex5 Device

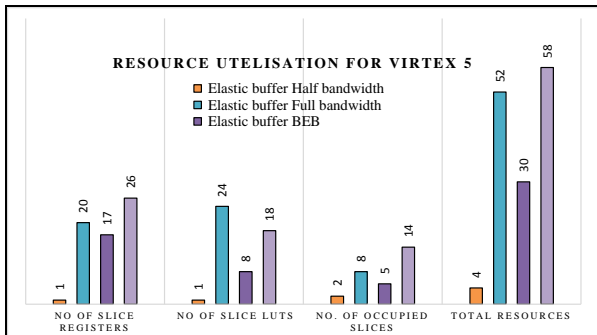


Fig.5. Plot of Total Resources used in Various Elastic Buffers on Virtex5 Device.

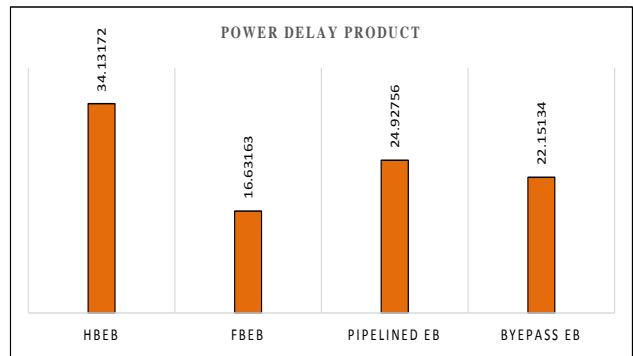


Fig.8. Plot of PDP in Various Elastic Buffers on Virtex5

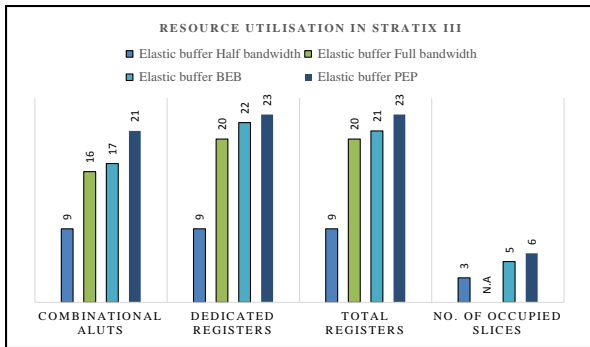


Fig.6. Plot of Total Resources used in Various Elastic Buffers on StratixIII

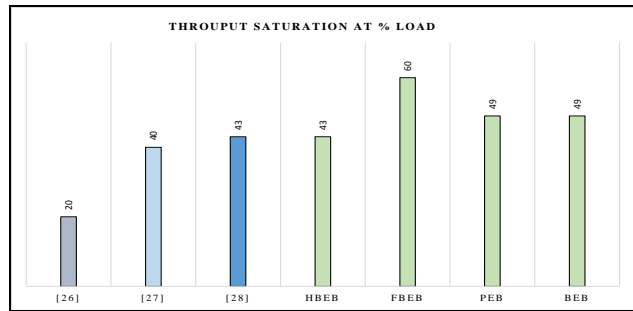


Fig.9. Plot of Throughput Saturation at Percent Load Injected

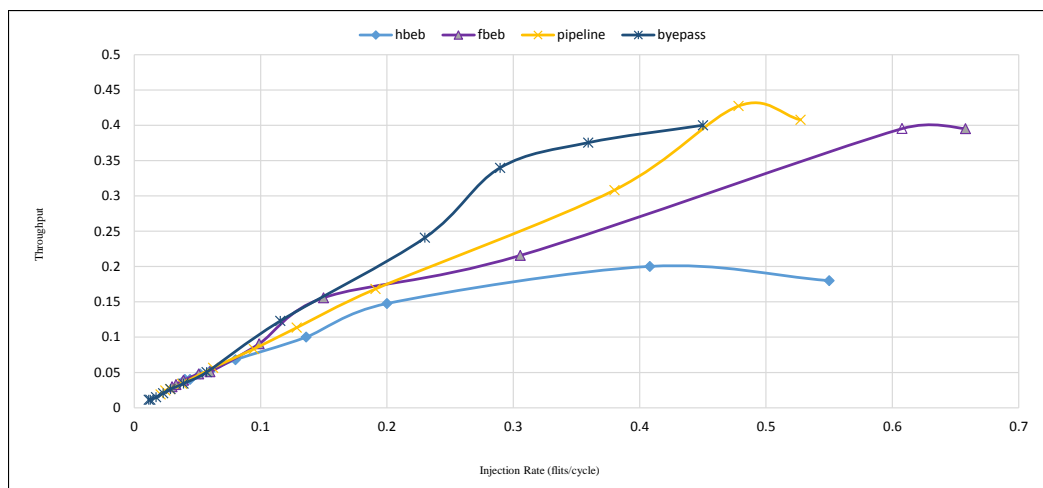


Fig.10. Plot of Throughput v/s Injection Rate

Figure 10 shows the relationship between throughput and traffic load injected into the input buffer. As seen from Figure FBEB, PEB and BEB provides a higher throughput than HBEB buffering policy. There exists a difference on the throughput with different priority. The throughput is higher with higher priority and it becomes a constant when traffic load arrives at a threshold. This also presents that PEB and BEB are better for higher throughput NOC architecture and can support real time communication.

## VI. CONCLUSIONS AND FUTURE WORK

This paper presented the performance analysis of various elastic buffering policies needed to design microarchitectural routers for NoC. The implementation was targeted for Virtex 5 FPGA device from Xilinx and Stratix III device from Altera. The resulting structures showed differences in the way of using resources available in the target FPGA device. The Bypass EB using the resources extensively has more PDP metric and larger ADP metric as compared to other elastic buffers. Pipelined EB uses lesser resources than other elastic buffering policies after HBEB and shows relatively reasonable PDP metric and lesser ADP metric with least ADP for HBEB. HBEB consumes least FPGA resources with least ADP but the PDP and Dynamic power dissipation of HBEB are highest and with least throughput parameters. The experimental results highlight the higher throughput capacity of the elastic buffering policies that are useful in designing simple Network on chip micro architectural routers. To sum up, a judicious trade-off between area, power and throughput parameters, and the intended application will determine the correct approach for adopting the correct elastic buffering policy for optimized design of an on-chip router. Since the throughput, ADP and the PDP of the design targeted for FPGA platforms are considered as important parameters, hence implementation of above the elastic buffering policy will be helpful for Network-on-chip (NoC) communication architecture design community. Further work is aimed to implement other buffering policies, alternative full throughput elastic buffers full generic elastic buffer and evaluate them for credit based flow control protocol used in NoC router communication with neighboring routers. Resource utilization and throughput for FPGA device would be evaluated by inserting pipelining units in the credit based flow control protocol, which would further affect the resource utilization and increase the throughput of the link.

## REFERENCES

- [1] B. Osterloh, H. Michalik, B. Fieth, K.Kotarowski, "SoCWire: A Network-on-Chip Approach for Reconfigurable System-on-Chip Designs in Space Applications," in proc of NASA/ESA Conference on Adaptive Hardware and Systems, pp 51-56, June 2008.
- [2] Abdelrasul Maher, R. Mohhamed, G. Victor., "Evaluation of The Scalability of Round Robin Arbiters for NoC Routers on FPGA," 7<sup>th</sup> International symposium on Embedded Multicore/Manycore System-on-chip, pp61-66, 2013.
- [3] Akram Ben Ahmaed, Abderazek Ben Abdallah, Kenichi, Kuroda, "Architecture and Design of Efficient 3D Network-on-Chip (3D NoC) for custom multicore SoC," in International conference on Broadband, Wireless Computing, communication and Application, FIT, Fukuoka, Japan, Nov 2010.
- [4] T. Anderson, S. Owicki, J. Saxe, and C. Thacker, "High speed switch scheduling for local area networks," ACM Trans. Comput. Syst., vol. 11, no. 4, pp. 319-352, Nov. 1993.
- [5] M. Karol and M. Hluchyj, "Queueing in high-performance packetswitching," IEEE J. Select. Areas Commun., vol. 6, pp. 1587-1597, Dec. 1988.
- [6] N. McKeown, V. Anantharam, and J. Walrand, "Achieving 100% throughput in an input-queued switch," in Proc. IEEE INFOCOM '96, San Francisco, CA, pp. 296-302.
- [7] Yuan-Ying Chang, Huang, Y.S.-C., Poremba, M. Narayanan, V. Yuan, Xie King, C., "Title TS-Router: On maximizing the Quality-of-Allocation in the On-Chip Network," in IEEE 19th International Symposium on High Performance Computer Architecture (HPCA2013), pp 390-399, Feb 2013.
- [8] B. Phanibhushana, K. Ganeshpure, S. Kundu, "Task model for on-chip communication infrastructure design for multicore systems," in proc of IEEE 29th International Conference on Computer Design (ICCD), pp 360-365, Oct 2011.
- [9] J. Guo, J. Yao, Laxmi Bhuyan, "An efficient packet scheduling algorithm in network processors," in proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies, pp 807- 818, March 2005.
- [10] William John Dally, Brain Towels, Principles and Practices of Interconnection Networks, 1st ed. Morgan Kaufmann publications, 2003.
- [11] N. McKeown, "Scheduling algorithms for input buffered cell switches," Ph.D thesis, University of California at Berkeley, 1995.
- [12] K. Lee, Se-je Lee, Hoi-jun Yoo, "A Distributed Crossbar Switch Scheduler for On-Chip Networks," in Custom Integrated Circuits Conference, 2003. Proceedings of the IEEE, pp 671-674, Sept. 2003.
- [13] P. Gupta, N. McKeown, "Designing and implementing a Fast Crossbar Scheduler," in proc of Micro, IEEE, vol 19, no 1, pp 20-28, Feb 1999
- [14] Kangmin Lee, Se-joong, Hoi-jun Yoo, "A distributed crossbar switch scheduler for On-Chip-Networks," in proceedings of IEEE Custom Integrated Circuits Conference, pp 671-674, 2003.
- [15] Nick McKeown, "The iSLIP Scheduling Algorithm for Input-Queued Switches" IEEE/ACM transactions on Networking, vol 7, no, 2, April 1999.
- [16] Jin Ouyang, Yuan Xie, "LOFT: A High Performance Network-on-Chip Providing Quality-of-Service Support," in 43rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO), pp 409-420, Dec 2010.
- [17] G. Jose, Delgado-Frias, B.R Girish, "A VLSI Crossbar switch with Wrapped Wave Front Arbitration," IEEE transaction on circuits and systems-I: Fundamental theory and applications, Vol 50, No 1, Jan 2003.
- [18] <http://www.xilinx.com>.
- [19] Dally, W.J., "Virtual-channel flow control," in Parallel and Distributed Systems, IEEE Transactions on, vol.3, no.2, pp.194-205, Mar 1992.



- [20] Micheliogiannakis, G.; Dally, W.J., "Elastic Buffer Flow Control for On-Chip Networks," in *Computers, IEEE Transactions on*, vol.62, no.2, pp.295-309, Feb. 2013.
- [21] G. Micheliogiannakis, J. Balfour, and W.J. Dally, "Elastic Buffer Flow Control for On-Chip Networks," *Proc. IEEE 15th Int'l Symp. High-Performance Computer Architecture (HPCA '09)*, pp. 151-162, 2009.
- [22] Seitanidis, I.; Psarras, A.; Chrysanthou, K.; Nicopoulos, C.; Dimitrakopoulos, G., "ElastiStore: Flexible Elastic Buffering for Virtual-Channel-Based Networks on Chip," in *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, vol.23, no.12, pp.3015-3028, Dec. 2015.
- [23] Marculescu, Radu, et al. "Outstanding research problems in NoC design: system, microarchitecture, and circuit perspectives." *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 28.1 (2009): 3-21.
- [24] Peh, Li-Shiuan, and William J. Dally. "Flit-reservation flow control." *High-Performance Computer Architecture, 2000. HPCA-6. Proceedings. Sixth International Symposium on*. IEEE, 2000.
- [25] Saastamoinen, I.; Alho, M.; Nurmi, J., "Buffer implementation for Proteo network-on-chip," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol.2, no., pp.II-113-II-116 vol.2, 25-28 May 2003.
- [26] Mello, Aline, et al. "Virtual channels in networks on chip: implementation and evaluation on hermes NoC." *Proceedings of the 18th annual symposium on Integrated circuits and system design. ACM, 2005*.
- [27] Gharan, M.O.; Khan, G.N., "A Novel Virtual Channel Implementation Technique for Multi-core On-chip Communication," in *Applications for Multi-Core Architectures (WAMCA), 2012 Third Workshop on*, vol., no., pp.36-41, 24-25 Oct. 2012.
- [28] Attia, B.; Abid, N.; Chouchen, W.; Zitouni, A.; Tourki, R., "VCRBCM: A low latency virtual channel router architecture based on blocking controller manger," in *Systems, Signals & Devices (SSD), 2013 10th International Multi-Conference on*, vol., no., pp.1-8, 18-21 March 2013.
- [29] Mukherjee, Shubhendu S., et al. "The Alpha 21364 network architecture." *Hot Interconnects 9, 2001.. IEEE, 2001*.
- [30] Peh, Li-Shiuan, and William J. Dally. "A delay model and speculative architecture for pipelined routers." *High-Performance Computer Architecture, 2001. HPCA. The Seventh International Symposium on*. IEEE, 2001.
- [31] Mullins, Robert, Andrew West, and Simon Moore. "Low-latency virtual-channel routers for on-chip networks." *ACM SIGARCH Computer Architecture News*. Vol. 32. No. 2. IEEE Computer Society, 2004.
- [32] Jerger, Natalie Enright, and Li-Shiuan Peh. "On-chip networks." *Synthesis Lectures on Computer Architecture* 4.1 (2009): 1-141.
- [33] Balfour, James, and William J. Dally. "Design tradeoffs for tiled CMP on-chip networks." *Proceedings of the 20th annual international conference on Supercomputing. ACM, 2006*.
- [34] Kim, John, James Balfour, and William Dally. "Flattened butterfly topology for on-chip networks." *Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture. IEEE Computer Society, 2007*.
- [35] Kumar, Amit, et al. "Express virtual channels: towards the ideal interconnection fabric." *ACM SIGARCH Computer Architecture News*. Vol. 35. No. 2. ACM, 2007.
- [36] Jain, Tushar NK, et al. "Asynchronous Bypass Channels for Multi-Synchronous NoCs: A Router Microarchitecture, Topology, and Routing Algorithm." *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on* 30.11 (2011): 1663-1676.
- [37] Ramanujam, R.S.; Soteriou, V.; Bill Lin; Li-Shiuan Peh, "Extending the Effective Throughput of NoCs With Distributed Shared-Buffer Routers," in *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, vol.30, no.4, pp.548-561, April 2011.
- [38] Serfozo, Richard. *Introduction to stochastic networks*. Vol. 44. Springer Science & Business Media, 2012.
- [39] So, Kut C., and E. Chin Ke-tsai. "Performance bounds on multiserver exponential tandem queues with finite buffers." *European journal of operational research* 63.3 (1992): 463-477.
- [40] Perros, H.G.; Altioik, Tayfur, "Approximate analysis of open networks of queues with blocking: Tandem configurations," in *Software Engineering, IEEE Transactions on*, vol.SE-12, no.3, pp.450-461, March 1986.
- [41] Buyukkoc, C., "An approximation method for feedforward queueing networks with finite buffers a manufacturing perspective," in *Robotics and Automation. Proceedings. 1986 IEEE International Conference on*, vol.3, no., pp.965-972, Apr 1986.
- [42] Bjerregaard, Tobias, and Jens Sparsø "Scheduling discipline for latency and bandwidth guarantees in asynchronous network-on-chip." *Asynchronous Circuits and Systems, 2005. ASYNC 2005. Proceedings. 11th IEEE International Symposium on*. IEEE, 2005.
- [43] Kleinrock, Leonard. *Theory, volume 1, Queueing systems*. Wiley-interscience, 1975.

### Authors' Profiles



**Liyaqat Nazir**, is presently a Ph.D. scholar in National Institute of Technology from India. He has received the B.Tech degree in Electronics and Communications Engineering from the Islamic University of Science and Technology, India, in 2011. He did his M.Tech degree in Communications and Information Technology from National Institute of Technology Srinagar, India in 2013. Currently he is a Ph.D. scholar in the department of Computer Science and Engineering, NIT, Srinagar. His main research interests include Network on chip, Digital VLSI design, Mixed signal design. Reconfigurable architectures, Architectural and technology dependent optimizations targeted for FPGA platforms, etc. He has many publications in the related field and is a Graduate student member of IEEE. He is also a lifetime member of IETE.

In recent years, Network on chip has been actively researched. Buffer management and buffering policies are the elementary problem in the applications.



**Roohie Naaz Mir** received B.E. (Hons) in Electrical Engineering from University of Kashmir (India), M.E. in Computer Science & Engineering from IISc Bangalore (India) in 1990 and Ph.D from University of Kashmir, (India) in 2005. She is currently a Professor in the department of Computer Science & Engineering at NIT Srinagar, India. She is the co-

author of many scientific publications in international journals and conferences. Her current research interests include Reconfigurable computing, security and routing in wireless ad-hoc networks, Sensor networks, High level computer architecture design Network on chip, Digital VLSI design,

Mixed signal design., Reconfigurable architectures, Architectural and technology dependent optimizations targeted for FPGA platforms, etc. she has authored many research articles in the related field and is a member of IEEE.

**How to cite this paper:** Liyaqat Nazir, Roohie Naaz Mir, "Realization of Efficient High Throughput Buffering Policies for Network on Chip Router", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.7, pp.61-70, 2016.DOI: 10.5815/ijcnis.2016.07.08