Evaluation of Efficient Elastic Buffers for Network on Chip Router

[1]Liyaqat Nazir
[1] Computer Science Engineering.
National Institute of Technology, Srinagar
J&K, India.
liyaqat_02phd13@nitsri.net

[2]Roohie Naaz Mir
[2]Computer Science Engineering.
National Institute of Technology, Srinagar
J&K, India.
naaz310@yahoo.co.in

Abstract—**The communication between processing elements are suffering challenges due to power, area and latency. Temporary flit storage during communication consumes maximum power of the whole power consumption of the chip. The majority of current NoCs consume 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 use of elastic buffers, input buffers are no longer required hence leading to a simplified router design. This paper evaluates some forms of the elastic buffers and presents the synthesis and implementation on FPGA platforms. The work will help NoC designers in suitable router implementation for their FPGA design The implementation targets Virtex5 FPGA and Stratix III device family.**

Keywords—**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 lot of attention by providing higher bandwidth and higher performance architectures for communication on chip. 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, torrous, 2D, 3D type of topologies as shown in figure 1. To achieve high performance 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 router throughput is determined by the critical path of the data path units in the router and the efficiency of control path units [5-8]. The data paths of the on-chip router comprises 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. 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 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 from its internal logic and dequeues the available data to the link, when the valid and ready signals are 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].

This work, therefore, focuses on the implementation and evaluation of with low port density half bandwidth and full bandwidth elastic buffers on reconfigurable platforms. The word length selected in this work is 8-bit vector. In this paper we carried out implementation and synthesis of various Half bandwidth and full bandwidth elastic buffers on Virtex 5 and stratix III FPGA platforms. The rest of the paper is organized as follows. Section 1 gives the introduction. Section 2 defines elastic channels and buffers briefly. Section 3 talks about operational details. Section 4 talks about synthesis and implementation. Section 5 provides the conclusion and future scope of the work carried out and finally references are listed in the end.

978-1-5386-0814-2/17/$31.00 ©2017 IEEE

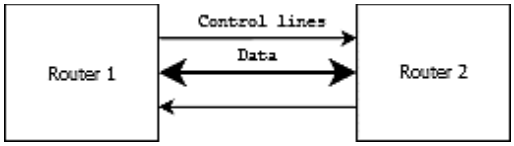

Figure1. Block diagram of NoC router communication.

II. ELASTIC CHANNELS AND BUFFERS

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 acknowledgement [20]. Block diagram of this protocol is shown in Fig 1.

III. OPERATIONAL DETAILS

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 2 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 a 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.

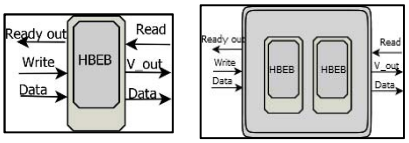

Figure2. Block diagram of (a)HBEB (b) FBEB

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 one 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 dequeue 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 table. 2.

C. pipelined and Byepass 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 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, 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.

IV. SYNTHESIS AND IMPLEMENTATION

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, 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 are 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 3 gives the comparison of the resource utilization of single slot Elastic buffer and two slot Elastic buffer implemented on Virtex5 FPGA device. The various resources utilized and the total resource utilization for the Virtex 5 device are plotted in figure 3. It can be seen that FBEB on account of high throughput utilizes the maximum on chip resources while as the HBEB being simple with throughput constraints utilizes the least on chip resources. Table 4 gives the comparison of the resource utilization of 1-slot EB and 2-slot EB implemented on Stratix III FPGA device. Table 4 provides a comparison of maximum achievable clock rates post implementation for a 8-bit data elastic buffer. From Table 4, we can see that the HBEB can operate at maximum frequency of 1027.908 MHz, while as FBEB operates at 890.27 MHz on Virtex5 FPGA., the corresponding operating frequencies on Stratix III device were measured as 1027.30 MHz and 1003.03 MHz for HBEB and FBEB respectively. Although FBEB having higher throughput, proves to be slowest in terms of maximum operating frequency of the elastic buffer structure. Finally static and dynamic power dissipation for for these two elastic buffers 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 (α), 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 5. 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 the HBEB and FBEB are plotted in figure 5. Similarly the area delay product is plotted in figure 6 below. It can be seen from figure 5 that the HBEB measures least area delay product. While as figure 6 depicts that EBEB measures least power delay product.

V. CONCLUSION AND FUTURE WORK

This paper carried out the performance analysis of HBEB and FBEB 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 Byepass EB uses the resources extensively has more power delay product and larger area delay product metric as compared to other elastic buffers. On the other hand pipelined-EB uses lesser resources than other elastic buffers and also shows relatively higher power delay product and least area delay product metric. To sum up, a judicious trade-off between area, power and throughput parameters, and the intended application will determine the correct approach for adopting correct elastic buffer for optimized design of an on-chip router. Since the area delay product and the power delay product of the design targeted for FPGA platforms are considered as important parameters. Hence implementation of above the elastic buffers will be helpful for Network-on-chip (NoC) communication architecture design community. We further intend to implement pipelined buffers, alternative full throughput elastic buffers, bye pass elastic buffers and full generic elastic buffer and evaluate them for credit based flow control protocol used in NoC router communication with neighboring routers. We further to intend evaluate resource utilization and throughput for FPGA device by inserting pipelining units in the credit based flow control protocol, this would further affect the resource utilization and increase the throughput of the link.

TABLE I. 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 II. 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 III. RESOURSE UTELIZATION COMPARISON FOR HBEB AND FBEB ON VIRTEX 5 FPGA.

On chip Resource	**Elastic buffer**			
	Half bandwidth	***Full bandwidth***	***Pipelined EB***	***Byepass 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 IV. RESOURSE UTELIZATION COMPARISON FOR HBEB AND FBEB ON STRATIX III FPGA.

On chip Resource	Elastic buffer			
	Half bandwidth	*Full bandwidth*	*Pipelined EB*	*Byepass 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 V. TIMING COMPARISON FOR HBEB AND FBEB ON VIRTEX 5 DEVICE

Timing Parameter	Elastic buffer			
	Half bandwidth	*Full bandwidth*	*Pipelined EB*	*Byepass 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 VI. POWER DISSIPATION FOR HBEB AND FBEB ON VIRTEX 5 DEVICE

FPGA Resource	Power dissipation (mW)			
	Encoding topology			
	Half bandwidth	*Full bandwidth*	*Pipelined EB*	*Byepass 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
Total	**3478.84**	**3473.83**	**3476.95**	**3478.46**

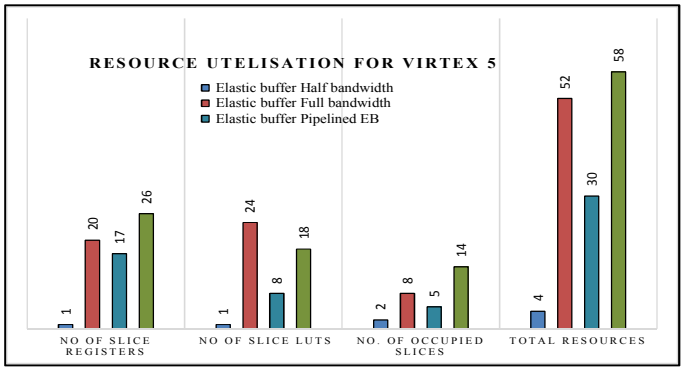

Figure 3. Plot of total resources used in various elastic buffers on Virtex5 device.

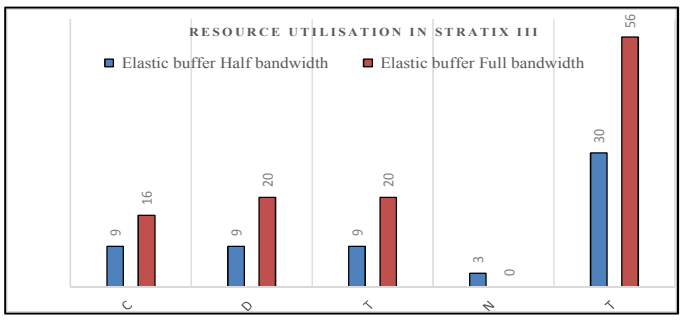

Figure 4. Plot of total resources used in various elastic buffers on StratixIII device

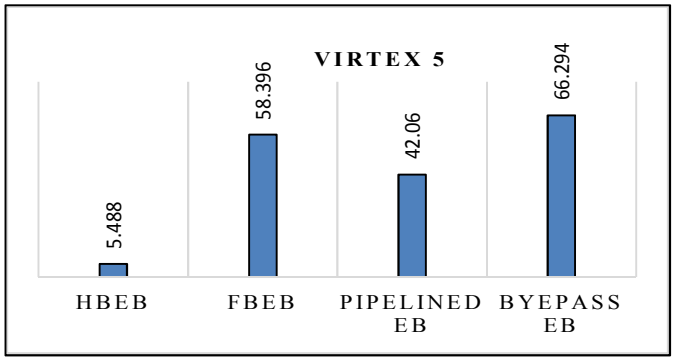

Figure 5. Plot of Area delay product in various elastic buffers on Virtex5 device.

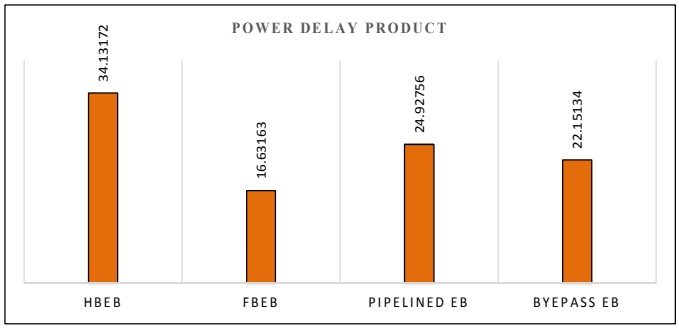

Figure 6. Plot of Power delay product in various elastic buffers on Virtex5 device.

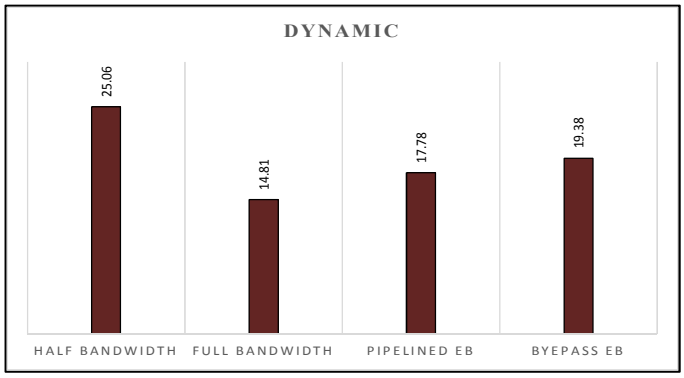

Figure 3. Plot of dynamic power used in various elastic buffers on Virtex5 device.

REFERENCES

[1] B. Osterloh, H. Michalik, B. Fiethe, 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,"7th 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 confrence 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, Ist ed. Morgan Kaufmann publications, 2003.

[11] N.Mckeown, "Scheduling algorithms for input buffered cell switches," Ph.D thesis, University of Calfornia at Berkely,1995.

[12] K. Lee, Se-jee 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 schedular 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 wih Wrapped Wave Front Arbitrarion," 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] Michelogiannakis, 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. Michelogiannakis, 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