Lasio 3D NoC Vertical Links Serialization: Evaluation of Latency and Buffer Occupancy

Yan Ghidini, Matheus Moreira, Lucas Brahm, Thais Webber, Ney Calazans, César Marcon
PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brazil
{yan.souza, matheus.moreira, lucas.brahm, thais.webber}@acad.pucrs.br, {ney.calazans, cesar.marcon}@pucrs.br

***Abstract*—Communication plays a crucial role in the design of high performance Multiprocessor Systems-on-Chip (MPSoC). Accordingly, Networks-on-Chip (NoC) have been successfully employed as a solution to deal with communication in complex MPSoCs. NoC-based architectures are characterized by various tradeoffs related to structural characteristics, performance specifications, and application demands. In new technologies, the relative values of wire delays and power consumption are increasing as the number of cores in 2D chips increase. The recent 3D IC technology applied to NoC architectures allows greater device integration and shorter interconnection links, which directly influences the communication performance. Through-Silicon Vias (TSVs) are used for the interconnection between vertical layers of a 3D IC. The drawback is that TSVs are usually very expensive in terms of silicon area, limiting their usage. This work explores the serialization of vertical links, employing a TSV multiplexing scheme for Lasio, a 3D mesh NoC. We implemented and analyzed the impact in network and application latency and in the occupancy of input buffers for a 4x4x4 mesh NoC with different multiplexing degrees, which imply different levels of TSV usage reduction and serialization. Results demonstrate that the proposed scheme allows reducing TSV usage with low performance overhead, pointing to potential benefits of the scheme in 3D NoC-based MPSoCs.**

Keywords—Networks-on-Chip, 3D IC technology, Through-Silicon Vias, TSV multiplexing.

I. INTRODUCTION

Networks-on-Chip (NoCs) are packet-based intrachip communication architectures for Multiprocessor System-on-Chip (MPSoC) design [1]. They improve scalability and throughput, and may reduce power when compared to traditional, shared bus architectures [2]. However, the increase in the number of cores on planar chip geometries may reduce efficiency due to the increase on the network diameter and end to end communication distance. One solution is extending 2D tile-based MPSoC architectures [3] to the 3D domain.

Indeed, 3D integration has attracted significant attention in recent years, as it provides new opportunities for chip architecture innovations. 3D NoCs are part of a cutting edge technology, integrating NoC and 3D technologies. Containing multiple silicon layers of active devices, 3D ICs allow better performance enhancements with less scaling concerns [4] [5]. Moreover, 3D NoCs have other potential benefits, including smaller chip paths, higher transistor density, shorter wiring delays, higher communication bandwidth and a considerable reduction in the length and number of global interconnections [6]. With contemporary technology, an interesting and viable solution in building 3D ICs is to stack several 2D IC layers together using Through-Silicon Vias (TSVs) vertical links [7].

The TSV technology is relatively recent. Data traffic through TSVs might become a bottleneck in 3D NoCs, because these vertical links have a much bigger footprint than ordinary NoC links and disparate electrical characteristics [8]. One technique that tends to reduce area and the number of vertical connections is communication serialization through TSV multiplexing. A TSV multiplexing scheme can be seen as an interesting approach that enables better design space exploration to reduce area of 3D NoCs. Hereupon, the technique can help coping with technological issues, especially the limited number of TSVs in a 3D system. However, this scheme clearly affects network and application latency. Therefore, some tradeoffs need to be evaluated to explore its advantages and drawbacks.

This paper proposes a vertical link serialization mechanism, which relies on TSV multiplexing for a 3D mesh NoC architecture called Lasio, an extension of the Hermes 2D NoC [9]. It provides an evaluation of network and application latency, as well as input ports buffers occupancy for several synthetic scenarios. The degrees of freedom considered in this evaluation are: *(i)* type of traffic– all-to-all and complement, *(ii)* injection rate, *(iii)* buffer depth, and *(iv)* packet size. Obtained results suggest that an adequate multiplexing scheme may reduce TSV usage without significant degradation in NoC overall performance.

The paper is organized as follows. Section II discusses related works on 3D IC technology and on the impact of TSV to 3D NoC design, pointing some TSV multiplexing structures. Section III describes the architecture of Lasio, which is the 3D mesh NoC used for our experimental purposes. Section IV presents the proposed serialization scheme. Section V discusses the experiments and evaluates the impact of serializing communication in vertical links under different scenarios. Finally, section VI draws some conclusions and directions for further work.

II. RELATED WORK

Current MPSoCs are normally implemented targeting 2D communication architectures. However, in the last few years, there has been a growing interest in 3D ICs from academia and industry to alleviate the interconnect bottleneck problem faced by 2D ICs. IBM [10] and Tezzaron [11] have recently presented promising preliminary results and test chips with 3D IC technology. Below follows a revision of some relevant related works on 3D integration.

978-1-4799-1132-5/13/$31.00 ©2013 IEEE

Dubois et al. [12] proposed a distributed routing algorithm for 3D integration of heterogeneous dies using TSVs. The work shows that the proposed routing algorithm presents acceptable network performance as the number of vertical connections decreases. Pasricha [13] has proposed the serialization of communication in TSVs, demonstrating that such serialization reduces the interconnect TSV footprint on each layer. This leads to a more efficient core layout across multiple layers, due to reduced congestion. Also, this author shows that a 4 to 1 serialization can save 70% of TSV area footprint at negligible performance and power overheads.

Ramanujan and Lin [3] proposed a layer-multiplexed (LM) architecture for 3D NoC that takes advantage of the short interlayer wiring delays enabled by 3D technology. LM replaces the one-layer-per-hop routing in a conventional 3D mesh by simpler vertical demultiplexing and multiplexing structures. Consequently, comparing LM with a conventional 3D mesh, the former consumes 27% less power, attains 14.5% higher average throughput, and achieves 33% lower worst-case hop count on a 4x4x4 topology.

Yong et al. [14] suggested a 3D NoC design technique to decrease the number of TSVs by grouping communication packets to prevent critical traffic and also to reduce the TSV overhead drawbacks, such as area and cost. Xu et al. [15] presented a study about the impact of TSV on 3D NoC design and analyzed both performance and manufacturing cost for different TSV quantities. Their analysis explored two designs with half and quarter connections between layers and showed that without their approach at least thousands of TSVs were required for full inter-layer connection in a 4×4×4 mesh NoC.

Liu et al. [8] propose a TSV multiplexing scheme among neighboring NoC routers. The idea is that adjacent NoC routers share vertical links based on the observation that TSVs utilization in a 3D symmetric mesh NoC is quite low, and adjacent routers rarely require data transmission in their vertical channels at the same time. The proposed solution can achieves much better performance in terms of network latency and throughput. Comparing [8] and the present work, the former proposes a multiplexing scheme that explores TSV multiplexing of four routers into a single one. This approach implies that the communication need of each router affects the communication rate of other routers in the same group. In our approach, serialization is performed for vertical links at each router. Thus, it does not directly affect other routers' communications.

Thus study focuses on a multiplexing scheme for vertical interconnections of the 3D mesh NoC called Lasio. We analyze the multiplexing impact on network and application latency as well as on the occupancy of input buffers on a 4×4×4 topology.

III. LASIO 3D NOC ARCHITECTURE

Lasio is a 3D mesh NoC architecture based on the Hermes 2D NoC [9]. Lasio employs the same switching and buffering mechanisms and resources of Hermes. It supports more ports at each router, using TSV interconnections between vertical router layers and a 7×7 crossbar to enable 3D communication, instead of the 5×5 crossbar of Hermes. The next Section details the Lasio characteristics.

A. Lasio Topology

Fig. 1 illustrates the Lasio 3D mesh NoC topology. Each layer can have multiple Processing Elements (PEs) such as memories and processors. Inter-layer communication channels are composed of TSVs that cut across thinned silicon substrates to build connectivity after die-bonding [16]. Lasio uses direct topology, where each tile contains a single PE connected to the NoC by a router local link. The direct topology eases router and PE placement, as well as building intra-network routing channels. This in turn simplifies building distributed routing algorithms. Besides, each Lasio router has a unique NoC address expressed in XYZ coordinates, and a different number of ports, depending on its position w.r.t. the NoC limits.

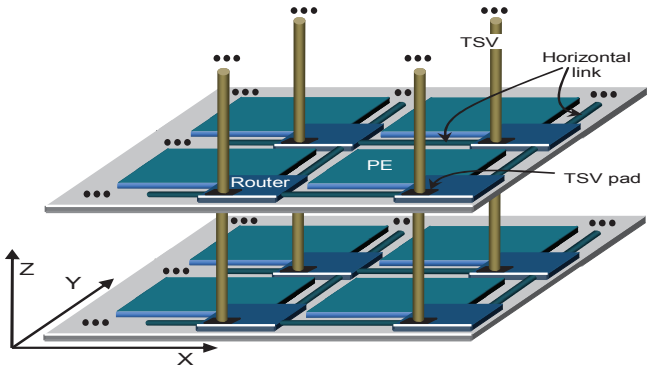

Fig. 1. A generic 3D mesh NoC.

B. Router Architecture and Interface

Fig. 2 illustrates the three basic modules that compose the Lasio router: (i) an input buffer for each one of the 7 ports that works as a circular FIFO; (ii) a switch control logic; and (iii) a 7x7 crossbar switch, responsible for ports interconnection.

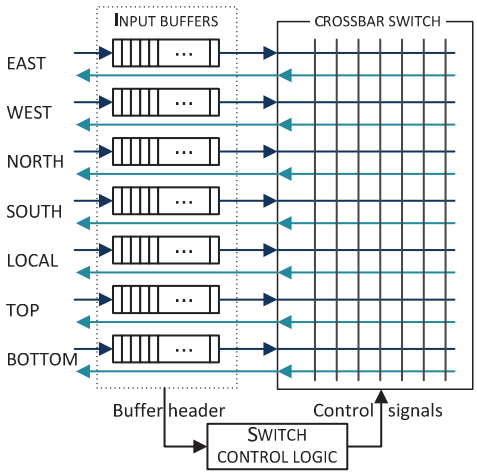

Fig. 2. Lasio router architecture.

When an input buffer receives the first flit of some packet, it sends this flit to the switch control logic that executes arbitration. If the incoming packet request is granted, it also performs the routing algorithm, connecting the input port data to the correct output port through the crossbar switch. If the chosen output port is busy, all subsequent flits are fed into the input buffer (as long as it has available room for them), and the request remains active until a connection with the output port is established.

Fig. 3 shows the structure of a bidirectional link between two routers located in the same plane. An output port comprises the following signals: (i) *clock_tx* that synchronizes data transmission; (ii) *tx* that controls data availability; (iii) *data_out*, a data bus; and (iv) *credit_i*, a control signal that indicates buffer availability. Conversely, an input port presents the following signals: (i) *clock_rx*; (ii) *rx*; (iii) *data_in*; and (iv) *credit_o*. These signals are the counterpart of the output port signals. Thus, each bidirectional link has 6 control signals and 2 data signals, and the NoC flit size defines the signals width.

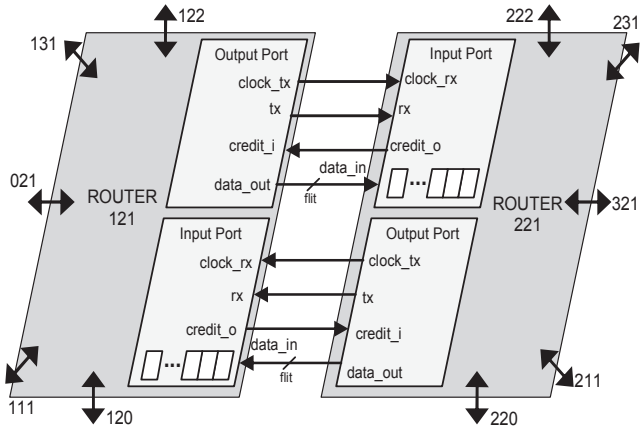

Fig. 3. Example of Lasio signals between routers 121 and 221.

C. Packet Structure

Fig. 4 depicts the Lasio packet structure, composed by an address field, a size field, and the variable size payload field. The address field has the target router address (i.e. XYZ coordinates of the target PE); the size field defines the amount of flits in the payload, the later containing the message.

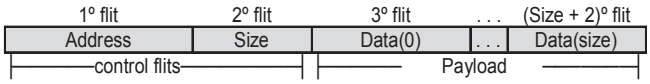

Fig. 4. Lasio packet structure.

D. Routing and Arbitration

Lasio implements the XYZ routing algorithm, which is deadlock free and enables small area overhead. When a router receives a header flit, the arbiter uses a dynamic rotating policy. In other words, arbitration is implemented using round robin. This method ensures that all incoming requests are processed, preventing starvation. If the routing algorithm is not able to establish a connection to the desired output port, the input port requires to the arbiter a new routing request.

E. Switching and Flow Control

Lasio implements wormhole switching method since: (i) this requires smaller buffers for storing data, and (ii) provides low-latency communication. The wormhole mode implies dividing packets into flits. The flit size for the Lasio infrastructure is customizable. Furthermore, Lasio uses credit-based flow control, an optimized communication mechanism, since it may take fewer clock cycles to perform a flit transmission (typically only one clock is required per flit). This method utilizes FIFO buffers with customizable size at the receiver input, and a return line (credit) to the transmitter, which informs if there is available space left in the destination input buffer. This information can be interpreted by the transmitter as a credit; consequently it just sends data as long as credit is available.

IV. Vertical Links Serialization Scheme

TSV is considered a promising and efficient technology for 3D integration, since it displays compatibility with standard CMOS processes [10]. However, TSV also brings some challenging problems that cannot be ignored. First, as Fig. 1 shows, TSV requires the provision of pads in each layer [13]. Analyzing TSV technology and planar metal wires, they are quite different from one another. For example, the former is not expected to scale with feature size [17], and the problem tends to become harsher as transistors and wires shrink. Second, a large number of distributed TSV pads across the whole network may aggravate routing congestion [13], which represents liability for high performance IC design. Finally, since current techniques of TSV fabrication still present relatively low yields, more TSVs may lead to lower yield [16].

Therefore, we propose a serialization scheme for communication on vertical links, based on TSV multiplexing, to minimize the above mentioned issues. Such schemes may be implemented at different levels depending on the employed routing algorithm and traffic scenario. Moreover, TSV sharing potentially leads to substantial area minimization, enabling better design space exploration.

Fig. 5 shows the scheme we propose between two vertically adjacent layers to serialize communication between routers.

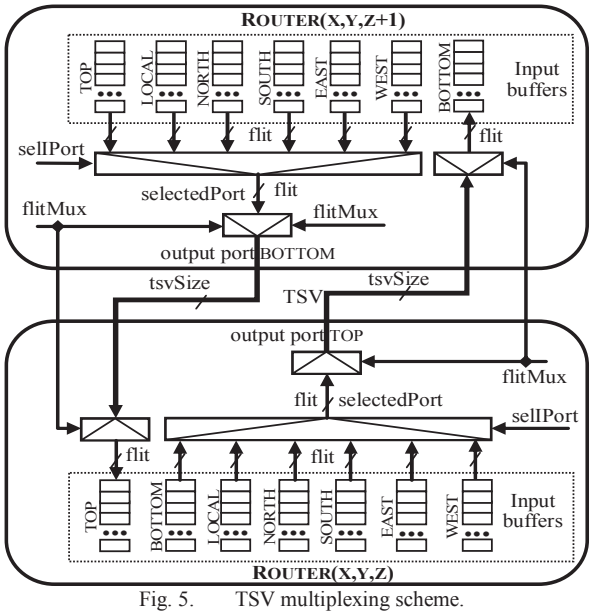

Fig. 5. TSV multiplexing scheme.

Each router has seven input buffers (with configurable depth) containing flits (with configurable size). TSVs multiplexing occurs on the bottom-up sense when data is coming from router (X, Y, Z) to router (X, Y, Z+1) or occurs on the top-down sense, when data is to be sent from router (X, Y, Z) to router (X, Y, Z-1), with $Z > 0$. For both senses data comes from one of six input ports from the router port destination. For example, in case of bottom-up multiplexing,

candidate input ports are NORTH, SOUTH, EAST, WEST, LOCAL and BOTTOM, whereas the TOP port should be the multiplexed data destination. Next, after the signal *selIPort* selects the input port, this signal is directed to *selectedPort*, a bus with (flit width) bits. TSV multiplexing is controlled by *flitMux*- a signal with $\log_2$ (flit width/tsvSize) bits - according to the number of TSV wires (i.e. tsvSize).

For instance, consider a NoC with 16-bit flits, 4-bit tsvSize and 2-bit *flitMux*, which enables to multiplex a flit in four steps. Once the signal is multiplexed, it is serially transmitted to the destination router through a TSV. At the target router, the TSV is demultiplexed by the demux circuit controlled by the same signal *flitMux*. Data is then deserialized and transmitted to the corresponding input buffer, which is the TOP buffer in the top-down multiplexing (or the BOTTOM buffer in case of bottom-up multiplexing). Accordingly, the number of multiplexed wires implies the number of steps (each requiring a single clock) needed to transmit all flit bits.

V. EXPERIMENTS AND DISCUSSION

We designed a simulation environment for the evaluation of the impact of serializing communication on vertical links of Lasio, as Fig. 6 depicts. Simulation consists of producer and consumer connected to each Local Port of each Lasio router in a 4×4×4 NoC instance. A consumer generates cycle accurate logs of input buffers occupancy and of the latency for each transmitted flit, measured as the difference between the time the consumer reads the data and the time the producer generates it. Moreover, producers allow selecting the injection rate, which controls packet insertion in the NoC. This was useful for evaluating the impact of the proposed scheme under different levels of traffic congestion. Our simulations employed injection rates of 1%, 2%, 5%, 10%, 15% and 20% for each scenario. These injection rates allow realistic comparisons, as many references point that real applications most often inject traffic in NoCs at rates below 15% [18].

Two synthetic applications were adopted as producers: (i) all-to-all and (ii) complement. In the former, all nodes send the same amount of data (uniform packet load) in a deterministic way to all other nodes, except to itself. This traffic model covers several traffic and blocking conditions, forcing that a large number of packets traverse the network simultaneously. This allows finding shortcomings of the communication infrastructure under analysis. The latter application generates and injects packets into the network in a similar way, but here each router sends packets to its *complementary router* (the complement function associates each router identifier to another router identifier, located in some way, in an opposite NoC position). The choice for the complement application derives from the fact that it uses vertical interconnections for all flit transmissions, allowing an efficient analysis of the proposed multiplexing scheme.

Because Lasio is parameterizable, variations in packet size and input buffers depth allow identifying the costs of multiplexing schemes for different network configurations. Flit size was assumed to be 16 bits. Albeit other flit sizes were also simulated, results varied only quantitatively and not qualitatively. Thus, these were omitted here. Simulations used five different buffer sizes: 4, 8, 16, 32 and 64. Packet sizes varied among 8, 16, 32 and 64, and 4 different multiplexing schemes were employed: 2/1, 4/1 and 8/1 multiplexing besides no multiplexing (for comparison). By varying each parameter at a time, as Fig. 6 details, we reach 5×4×4 = 80 different configurations of Lasio. All generated producer configurations for each Lasio instance combine to give 80 × 6 injection rates × 2 applications = 960 different simulations. The entire environment was described in VHDL and run with the Modelsim simulator. After simulation, an in-house tool analyzed data generated by the consumer and produced charts representing (i) average and worst case of router input buffers occupancy, (ii) average, worst and best case of network latency and (iii) total application latency.

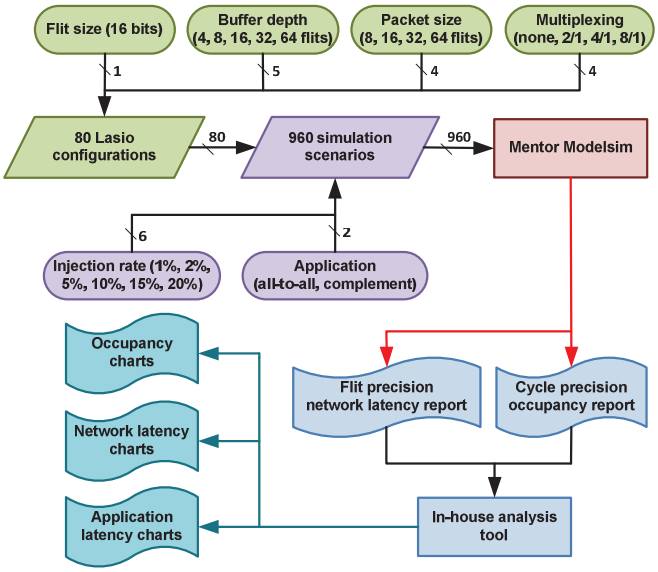

Fig. 6. Experimental setup.

Due to the huge amount of data, we chose a selection of charts to present here. We present only results for the complement traffic, since this produced the worst case for the proposed scheme. Fig. 7 shows average latency results, when varying the number of multiplexers as a function of flit size, injection rate and buffer size. The first line of charts represents the *measured latency* (in clock cycles) and the second line represents a *relative gain* (in percentage). This relative gain was measured as the expected losses using the multiplexing scheme divided by the measured latency overhead. For instance assume an 8/1 MUX. It is expected an 8 times bigger average latency, which is divided by the measured latency. In this way it is possible to evaluate the trade-offs of using multiplexing schemes.

Fig. 7(a) shows the impact of the packet size in the average network latency. The presented results are for a fixed injection rate of 10% and 8-flit buffers. Fig. 7(b) shows the impact of the injection rate, with fixed packet size (i.e. 16 flits) and 8-flit buffers, and Fig. 7(c) shows the effect of buffers depth, for a fixed injection rate of 10% and packet size 16 flits. As the first line charts of Fig. 7 show, the bigger the injection rate, the packet size or the buffers depth, the bigger is the average network latency. This is valid for all scenarios, independent of the multiplexing scheme.

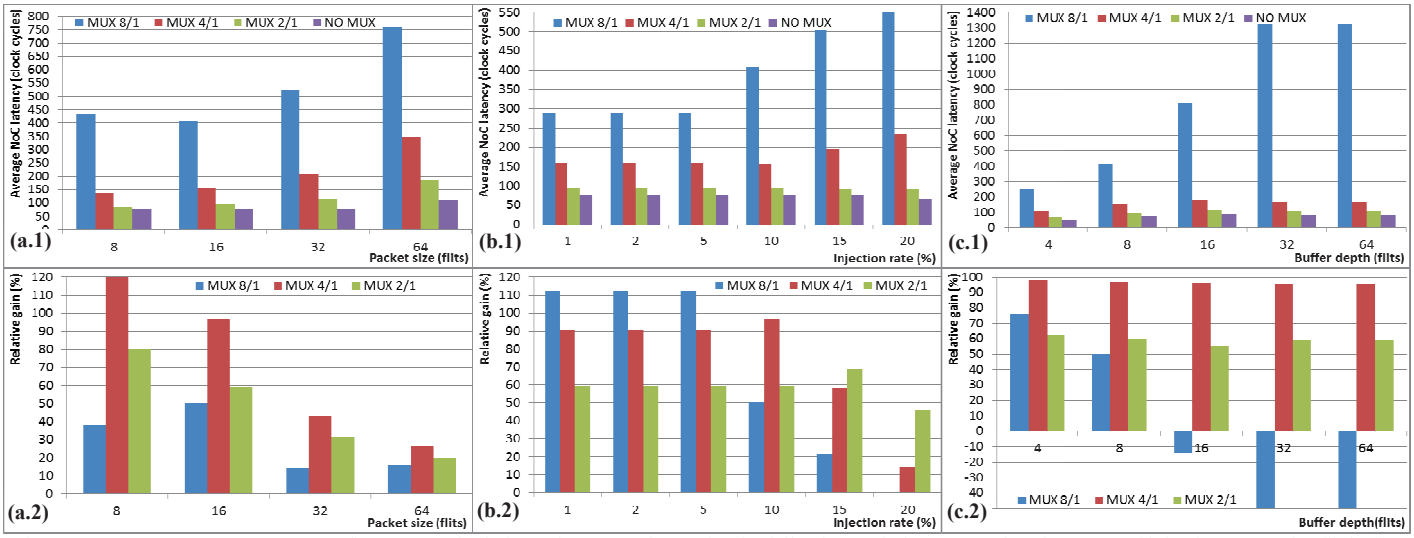

Fig. 7. Average **network latency** (first row) and relative gains (second row) for the following variations: (a)packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).

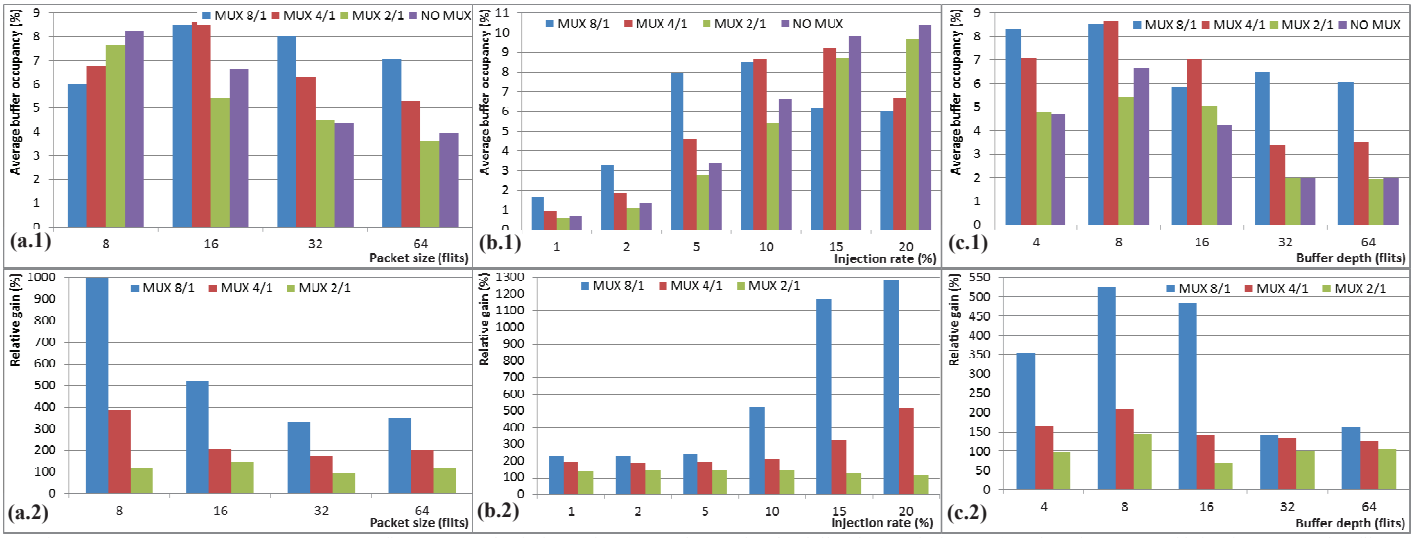

Fig. 8. Average **buffers occupancy**(first row) and relative gains (second row) for the following variations: (a)packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).

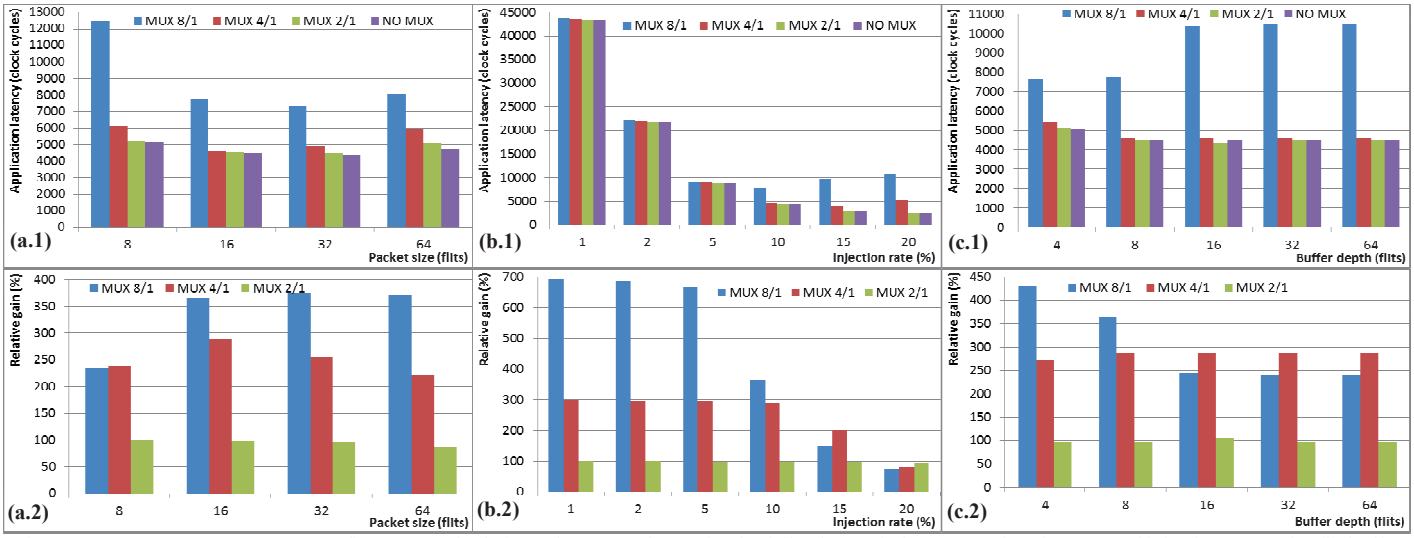

Fig. 9. **Total application latency**(first row) and relative gains (second row) for the following variations: (a)packet size (10% of injection rate and 8-flit buffer depth); (b) injection rate (16-flit packet size and 8-flit buffer depth); (c) buffer depth (10% of injection rate and 16-flit packet size).

Fig. 7(a) shows that for all packet sizes albeit MUX8/1 Lasio requires 8 times less TSVs than an equivalent NoC with no serialization, it does not increase 8 times the average network latency. In fact, the serialization scheme slightly increases the network latency (e.g. for 8-flit packet size NoC latency increases less than 4.8 times).Therefore, we can assume a relative gain of roughly 38% as showed in the chart of the second row in Fig. 7(a.2). Accordingly, all charts of the second row of Fig. 7, i.e. Fig. 7(a.2), Fig. 7(b.2) and Fig. 7(c.2), present relative gains. These values are useful to measure the cost-benefit relationship of the proposed scheme. As charts show, MUX2/1 and MUX4/1 always present positive relative gains, independent on the scenario, which indicates that the serialization scheme, for these multiplexing levels, can help coping with technological problems without worsening average network latency. On the other hand, the MUX8/1 presented negative gains for big buffers and for big injection rates, which suggests the lack of scalability of this scheme. In this way, it is possible to identify a saturation point for serialization and TSVs multiplexing for the Lasio NoC. Similar results are expectable for other NoC architectures.

Similar results are obtained when measuring the impact of the variations in the average buffer occupancy. All results display average occupancy of all top ports buffers of the NoC for each scenario. Albeit values for the bottom ports are also available, results varied only quantitatively and were omitted. As the first row of Fig. 8 shows, increasing the injection rate, the buffer depth or the packet size does not have a predictable impact in buffer occupancy. This is explained by the fact that such variations have an impact in other variables such as network and application latency, which end up affecting buffer occupancy figures.

It is important to notice that bigger injection rates applied to different multiplexing schemes do not meaningfully increase buffer occupancy, suggesting that occupancy is not proportionally increased with TSV reduction. Moreover, in experiments with the MUX8/1 scheme, the bigger the injection rate is, the bigger is the relative gain, indicating reduced occupancy figures. On the other hand, as Fig. 7 and Fig. 9 show, the average network latency and total application latency increase for this implementation, which leads to the misleading impression that buffer occupancy reduction is a positive result. In fact, this result only supports the statement that there is saturation point for the advantages of employing serialization schemes.

Finally, the impact of the simulated scenarios in the total application latency is summarized by Fig. 9. These values were measured as the total time for executing the application for each scenario. As Fig. 9(a.1), Fig. 9(b.1) and Fig. 9(c.1) show, for schemes MUX 2/1 and 4/1 , independent on the injection rate, on the buffer size and on the packet size, the total application time is always the same, or very similar to the scenario without serialization. For most cases of the MUX 8/1 scheme, total application latency increases, but the improvements on number of TSVs reduced are more substantial. In fact, as the second row of Fig. 9 shows, the relative gains for all multiplexing levels are elevated, indicating that the advantages of serializing vertical links overcome the drawbacks that could arise in system performance.

VI. CONCLUSIONS

This paper proposed and evaluated a serialization scheme for vertical links of the Lasio 3D NoC by multiplexing TSVs. The obtained results suggest that, albeit there are some losses in network latency, the proposed scheme still presents significant relative gains, as its improvements in terms of TSV reduction are more substantial than the reported losses. Also, lower losses were observed for schemes MUX 4/1 and 2/1, whereas for 8/1 schemes these losses were bigger, indicating a saturation point for the benefits of employing serialization. In this way, the reported results suggest that the proposed scheme is well suited to cope with 3D IC era challenges. Future works include the evaluation of other scenarios, with new applications that force higher congestion levels in the network and also other configurations of Lasio, including bigger flit size and network dimensions. It is also a future work the evaluation of the impact of serialization in different routing algorithms and in application mapping.

ACKNOWLEDGEMENTS

This work is partially funded by FAPERGS (under grants PqG 12/1777-4 and Docfix SPI n.2843-25.51/12-3), and by the CNPq (under grants 310864/2011-9, 556779/2009-6 and 552699/2011-0).

REFERENCES

[1] F. Fu et al., “A NoC Performance Evaluation Platform Supporting Designs at Multiple Levels of Abstraction”, in: ICIEA, 2009.
[2] A. Jantsch et al., “Network on Chip”, Kluwer Academic Publishers, 2003.
[3] R. S. Ramanujan, B. Lin, “A Layer-Multiplexed 3D On-Chip Network Architecture”, IEEE Embedded Systems Letters, 1(2), 2009.
[4] A. Rahmani et al., “LastZ: An Ultra Optimized 3D Networks-on-Chip Architecture”, in: EUROMICRO-DSD, 2011.
[5] A-M. Rahmani et al., “Research and Practices on 3D Networks-on-Chip Architectures”, in: NORCHIP, 2010.
[6] V. F. Pavlidis, E. G. Friedman, “3-D Topologies for Networks-on-Chip”, IEEE Trans. on VLSI Systems, 15(10), Oct 2007.
[7] V. F. Pavlidis, E. G. Friedman, “Three-dimensional Integrated Circuit Design”, Morgan Kaufmann, 2008.
[8] C. Liu et al. “Vertical Interconnects Squeezing in Symmetric 3D Mesh Network-on-Chip”, in: ASP-DAC, 2011.
[9] F. Moraes et. al., “HERMES: an infrastructure for low area overhead packet-switching networks on chip”, Integration the VLSI Journal, 38(1), 2004.
[10] K. Bernstein, et al., “Interconnects in the Third Dimension: Design Challenges for 3D ICs”, in: DAC, 2007.
[11] R. S. Patti, “Three-Dimensional Integrated Circuits and the Future of System-on-Chip Designs”, Proceedings of the IEEE, 94(6), June 2006. 2006.
[12] F. Dubois et al. “Elevator-First: A Deadlock-Free Distributed Routing Algorithm for Vertically Partially Connected 3D-NoCs”. IEEE Trans. on Computers, 62(3), 2013.
[13] S. Pasricha, “Exploring Serial Vertical Interconnects for 3D ICs”, in: DAC, 2009.
[14] Yong et al. “3D Network-on-Chip System Communication Using Minimum Number of TSVs”, in: ICTC, 2011.
[15] T. Xu et al., “A Study of Through Silicon Via Impact to 3D Network-on-Chip Design”, in: ICEIE, 2010.
[16] I. Loi et al., “A low-overhead fault tolerance scheme for TSV-based 3D network on chip links”, in: ICCAD, 2008.
[17] S. Das et al., “Technology, performance, and computer-aided design of three-dimensional integrated circuits”, in: ISPD, 2004.
[18] W. J. Dally, B. Towles, “Principles and Practices on Interconnection Networks”, Elsevier, 2004.