OcNoC: Efficient One-cycle Router Implementation for 3D Mesh Network-on-Chip

Ramon Fernandes, Lucas Brahm, Thais Webber, Rodrigo Cataldo, Letícia B. Poehls, César Marcon

PUCRS - Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, Brazil – 90619-900

{ramon.fernandes, lucas.brahm, thais.webber, rodrigo.cataldo}@acad.pucrs.br {leticia.poehls, cesar.marcon}@pucrs.br

Abstract —The overall system-on-chip performance depends on the network architecture, whose communication latency significantly impacts on the application performance. The challenge for on-chip networks is reducing costs while providing high performance such as low latency and high throughput. One alternative to achieve such goals is to implement efficient router architectures capable of fast packet switching and routing for parallel and scalable Networks-on-Chip (NoCs). We propose a single cycle router implementation for 3D Mesh NoCs with two arbitration approaches. Our evaluations show that the proposed one-cycle router can reduce network latency up to 57% and application latency up to 67%, when compared to multistage routers. This improvement comes with minimal silicon area overhead when compared to baseline router microarchitecture, while still maintaining short critical paths.

Keywords - 3D mesh NoC, routing, arbitration, throughput, latency, area consumption.

I. INTRODUCTION

The next generation of multiprocessor will encompass hundreds of integrated processors on a single chip with the promise of high throughput and low latency. However, as the number of processors increases, intrachip communication becomes a bottleneck in terms of power consumption and performance [1].

Network-on-Chip (NoC) has emerged as an excellent communication architecture for complex multiprocessor systems since it offers better performance, throughput and scalability compared to shared bus systems, and has high degree of data transmission parallelism [2].

Two-Dimensional (2D) NoCs interconnect routers and Processing Elements (PEs) in the same plane. Therefore, delays generated by communication over long wires and power consumption are evident problems in 2D NoCs with many and large PEs. This problem motivates the research for new models and communication topologies. The inclusion of the third dimension (i.e. 3D NoCs) reduces communication distances and the number of hops required to reach destinations, consequently reducing network latency and even the average power consumption [3].

Routers implement deterministic or adaptive algorithms [4], which might encompass Quality of Service (QoS) policies and fault tolerance strategies. As router complexity increases, it is natural to expect an additional processing overhead, negatively impacting the overall NoC performance, as well as increasing the NoC's area and power requirements. Therefore, efficient router implementation becomes a key aspect for NoC design.

This paper introduces OcNoC, which is a one-cycle 3D Mesh NoC based on the Lasio NoC [5]. The OcNoC's router may be implemented according to two different arbitration models. Wormhole switching is performed by a single stage mechanism, capable of evaluating port availability in a centralized or distributed fashion, using variable length buffers for handling traffic contention situations.

This paper is organized as follows: Section II discusses some related works. In Section III, we present OcNoC's architecture, emphasizing its differences with respect to a Lasio NoC. Section IV presents the experimental setup elaborated for evaluating OcNoC performance and design characteristics. Section V discusses the simulation results. Lastly, Section VI presents our conclusions and future works.

II. RELATED WORK

There have been significant researches aiming to reduce NoC communication latencies, such as designing new topologies and developing more efficient routers. In the following we present selected work whose major goal is to provide higher efficiency through router and buffers optimizations. Due to the fact that the presented approach focuses on router microarchitecture to improve arbitration cycle and enable faster route decisions, many of the works presented in this chapter differ in their adopted strategies. It is important though to point out that their overall aims, including latency reduction, throughput improvement and low area overhead, match.

Gomez et al. [6] propose a new switching technique called Blind Packet Switching (BPS) focused on increasing network frequency, reducing area and energy consumption. BPS replaces switch port buffers with single conventional latches, thus network cycles can be reduced, which decreases packet latency and alleviating the critical path, since the switch frequency can be doubled. Authors presented results for 4×4, 8×8, 10×10 mesh NoCs evaluating average packet Latency versus throughput, comparing wormhole and BPS. According to the authors, BPS outperforms wormhole in all cases and the evaluation has shown low occurrence of BPS drawbacks (packet reinjection, nack packets, and out-of-order delivery).

Kim [7] proposes a low-cost dimension-sliced router microarchitecture consisting of just pipeline registers and MUXs, partitioning the crossbar, including prioritized switch arbitration, and reducing the amount of buffers. Assuming a 2D mesh topology, the author introduces intermediate buffers internally to the router. Synthetic workload comparison (uniform random, bit complement, and transpose traffic patterns) using closed-

1063-9667/15 $31.00 © 2015 IEEE
DOI 10.1109/VLSID.2015.23

loop simulation shows an area reduction by 37% and power consumption savings of 45% compared to a baseline router microarchitecture that achieves similar throughput.

Nguyen and Oyanagi [8] propose a low latency router architecture, which utilizes Virtual Output Queuing (VOQ) to shorten the processing time of packet transfers. The simulations are carried out on a 4×4 2D mesh network developed in Verilog. Besides, a dedicated Virtual Channel (VC) router with baseline and look-ahead speculative architecture is implemented for comparison. All simulations are performed under uniform traffic and packet length is fixed to 5-flit. The authors evaluated the router in terms of communication latency as a function of the injection rate, throughput, and hardware amount. The proposed design with single VOQ architecture reduces area cost by 67.3% and communication latency by 25% when compared to the look-ahead speculative VC router.

Lai et al. [9] propose a single-cycle router architecture with wing channel, which enables the forwarding of the incoming packets to free ports. Their 2D mesh topology reduces the communication latency by more than 45%. The architecture supports different routing schemes under deterministic and adaptive traffic patterns, and results showed 14% of enhanced throughput, area overhead around 8%, and power consumption savings up to 7.8% in consequence of less arbitration activities.

Chen et al. [10] present a single-cycle output buffered router based on layered switching, which implements wormhole on top of virtual cut-through switching for 2D mesh/torus topologies. Their router reduces the area by 11% compared to an input virtual-channel router with the same buffer capacity. Moreover, the layered switching achieves up to 36.9% latency reduction for 12-flit packets under uniform random traffic with an injection rate of 0.5 flit/cycle/node.

Hassan and Yalamanchili [11] propose an energy-efficient router architecture, containing centralized and elastic buffers for link optimization, which is able to produce single cycle operation. Tests, using 4 synthetic traffic patterns, were performed with mesh, torus and generalized hypercube topologies for both 2D and 3D networks. The comparisons show an average improvement in throughput and latency for some benchmarks configured in a 2D Mesh topology.

Jonna et al. [12] propose a single cycle deflection router, which is minimally buffered (MinBSD). The router reduces the critical path latency and ensures smooth flow of flits through the router pipeline, performing overlapped execution of independent operations. Experimental results on an 8x8 mesh network with synthetic traffic showed that MinBSD reduces the average flit latency, area and power consumption when compared to the existing state-of-the-art minimally buffered deflection routers.

Our work uses combinational logic circuitry to implement the routing algorithm, due do its simplicity, as well as two arbitration methods for packet switching: centralized arbitration, which sequentially evaluates switch requests, and distributed arbitration, which evaluates switch requests in parallel enabling more efficient router operation.

III. OcNoC's Architecture

This section describes the main characteristics of the OcNoC's architecture, which is a 3D mesh NoC based on the Lasio NoC [5]. The OcNoC's router implements the deterministic XYZ routing algorithm with wormhole switching, aiming to reduce the overall network latency and buffers depth.

A. Network Topology and Addressing Model

Each router of the 3D mesh network contains seven distinct ports: six of these ports are used for connecting other routers, and the remaining port is employed for local PE interconnection. While horizontal links use standard wires for router interconnection, as shown in Figure 1, vertical links employ TSV technology. Independent on the NoC layer, every router has the same architecture and size, and independent on link connection (i.e. horizontal or vertical), flits are transmitted with the same quantity of clock cycles. We assume that each hop between routers has the same cost.

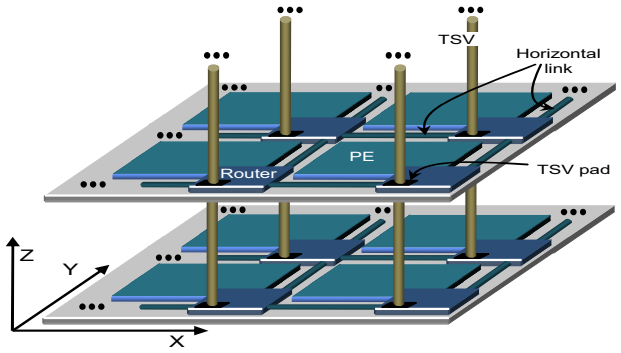


Figure 1 – 3D OcNoC topology, with TSV technology used for vertical links.

The OcNoC supports any 3D mesh configuration, which translates into 2x2x2, 4x4x4 or 4x4x2 layouts, for example, limited only by its addressing capabilities.

PEs are uniquely identified by their corresponding routers in the OcNoC architecture. Considering that each router in a plane is identified by a pair of X and Y coordinates, and that each plane is characterized by a Z coordinate, PE addressing comprises of X, Y and Z coordinates for identification.

Figure 2 shows OcNoC's packet structure, expressed in flits. The first flit contains the destination address (i.e. the XYZ coordinates of the target PE), to which routers must forward a given packet. The second flit is the payload packet length, while subsequent flits contain the payload itself.

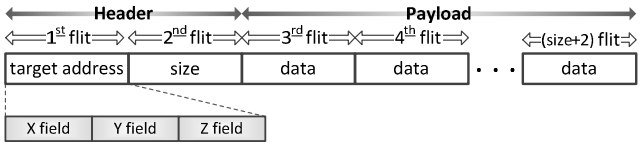


Figure 2 – OcNoC packet structure.

B. Router Interface

Figure 3 shows the main signals of a router's bidirectional link, which enable full-duplex communication, using a credit-based protocol. Each input port has three control signals: (i) *clockRx* for data synchronization; (ii) *rx* for data availability signaling; (iii) *creditOut* for indicating buffer availability; and

the data signal *dataIn*, which is a flit-bit size bus for receiving data. The output port employs the counterpart control/data signals: (i) *clockTx*; (ii) *tx*; (iii) *creditIn*; and *dataIn*.

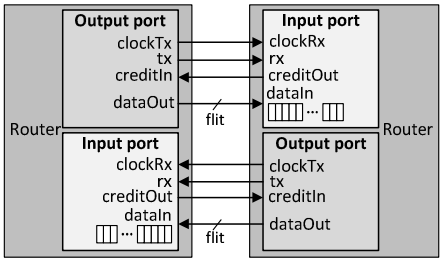
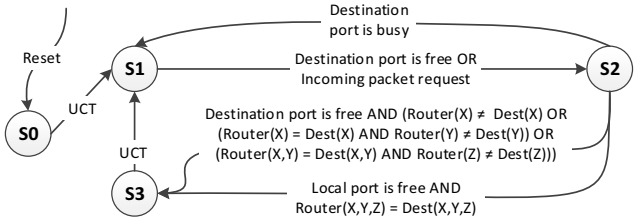


Figure 3 – Control and data signals of OcNoC ports.

C. *Router Architecture and Arbitration Mechanisms*

Figure 4 presents the arbitration and routing mechanism of Lasio's router through a Finite State Machine (FSM) in order to show OcNoC improvements.



Legend: UCT – Unconditional transition; Dest(...) – destination address

Figure 4 – Arbitration and routing mechanism of Lasio's router.

The FSM consists of four-stage machine, which accounts for the five required cycles for routing a packet in a router:

- **S0** (*Initializing state*) – the router passes once by S0 to perform initializing procedures (e.g., to set some register status), then FSM switches to state **S1** after a clock cycle;
- **S1** (*Waiting state*) – the FSM remains waiting for incoming packets through any of the input ports on the router, or FSM deals with packets that remain waiting for the release of the destination port (i.e., the other packet using the same destination port release it). At this point of time the FSM switches to state **S2**;
- **S2** (*Verifying states*) – **S2** is a composition of two states implemented in two clock cycles that are responsible for verifying the packet destination address against the router address and the corresponding destination port. If the destination port is free, the FSM finishes the arbitration and switches to **S3**. On the other hand, when destination port is busy, the FSM switches to **S1** for future arbitration (i.e. reswitching);
- **S3** (*Ending state*) - S3 is also implemented with two states. One is responsible for all flits delivering through the selected port, and the other one finishes the switching process by freeing the incoming data port. After that, the FSM switches to state **S1** to process further switching and routing requests.

Routing and arbitration spent, in the worst case, 5 clock cycles to switch a requesting packet and 3 additional cycles for every re-switching due to network congestion (states **S1**-**S2**).

The OcNoC implements a single cycle routing mechanism capable of evaluating packet destination and port availability through combinational logic, using two arbitration approaches between which has to be chosen during design time: (i) centralized and (ii) distributed.

In centralized arbitration the single module *Arbitration* is responsible for evaluating switching requests from input to destination ports, as shown in Figure 5. If the request is granted, then packets from the input port are routed to its destination in a single clock cycle.

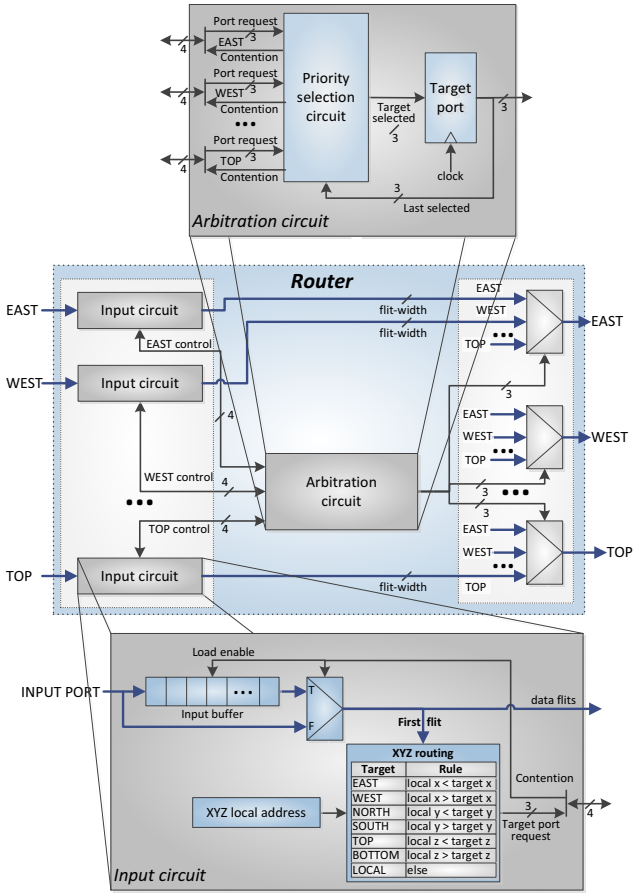


Figure 5 – Centralized arbitration scheme.

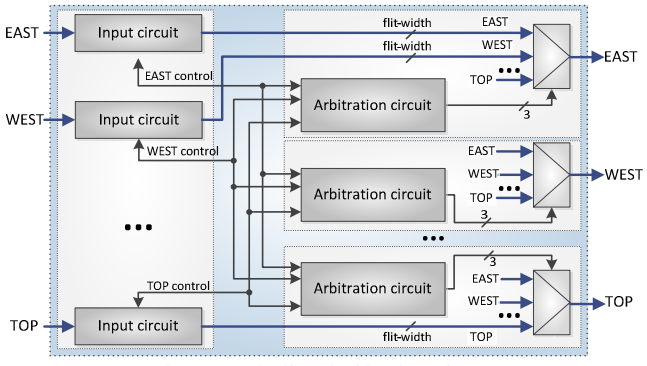


Figure 6 – Distributed arbitration scheme.

In cases where more than a single switching request is made, input ports are, based on Round Robin algorithm, evaluated in

series. This means that an input port, assuming that contention does not occur, might have to wait up to seven clock cycles to forward its packets to its destination. We propose a distributed arbitration mechanism in order to optimize this process, as shown in Figure 6. Each input port encompasses a dedicated unit for evaluating switch requests, so that waiting is now relative only to destination availability.

Both approaches require a rotation algorithm to guarantee fairness in the switching process. Supposing that the last port that was successfully granted switching permission was *North*, then it would be unfair to elect this same port for the next switching request before evaluating other requests.

Therefore, each port is given a certain priority, which itself is based on the previously selected port. Table 1 illustrates the rotation algorithm, where lower values represent higher priority. This table describes the functionality of the *Priority Selection Circuit*.

TABLE 1 – PRIORITY TABLE FOR SWITCHING ARBITRATION. LOWER VALUES EXPRESS A HIGHER PRIORITY.

		Last selected port						
		EAST	WEST	NORTH	SOUTH	LOCAL	BOTTOM	TOP
Next port	EAST	6	5	4	3	2	1	0
	WEST	0	6	5	4	3	2	1
	NORTH	1	0	6	5	4	3	2
	SOUTH	2	1	0	6	5	4	3
	LOCAL	3	2	1	0	6	5	4
	BOTTOM	4	3	2	1	0	6	5
	TOP	5	4	3	2	1	0	6

Considering that the *North* port has successfully forwarded its data to a destination in the previous routing cycle, it is now given a priority value 6, placing it last from other ports for future switching requests.

IV. EXPERIMENTAL SETUP

This work consists of several experiments comparing Lasio and OcNoC Centralized/Distributed within the following setup parameters: (i) 3-flit packet size, which is the minimum size of packet accepted by all NoC employed in this work, and 16-flit packet size; (ii) 8-flits depth buffers; (iii) 4×4×4 NoC mesh topology; (iv) all-to-all, complement and dataflow traffic scenarios; (v) eleven packet injection rates. Additionally, all experiments employs 16-bit of flit size. Figure 7 illustrates the performed experiments and tools used in this work.

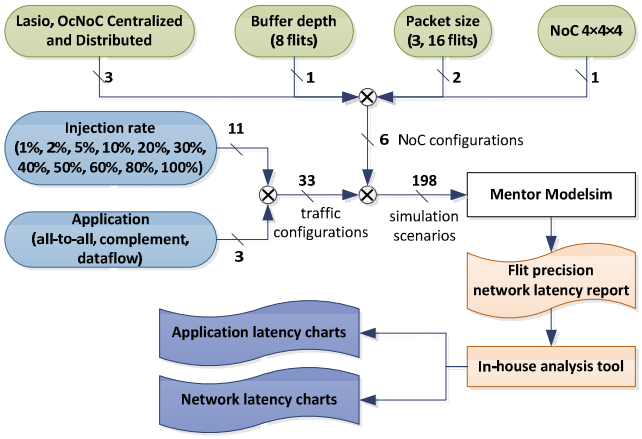


Figure 7 – Experimental Setup.

We employ three types of synthetic traffic in order to explore different aspects of each communication architecture, such as capacities in transport or switch packets.

In *complement* traffic, each PE sends traffic to its complementary PE in the NoC addressing model. Figure 8 shows the traffic pattern, where the first PE (000) sends data to the last PE (212), while the second PE (100) sends data to the penultimate PE (112). This traffic aims to congest a set of specific paths, since the traffic is constantly applied through a sequence of packets until application data are available.

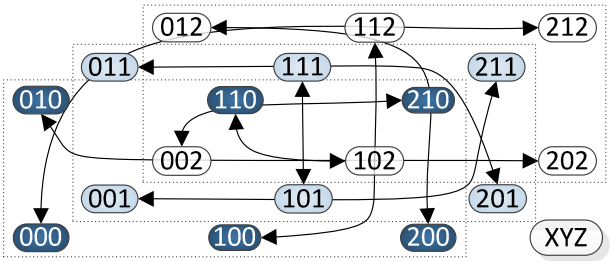


Figure 8 – Example of complement traffic pattern in a 3×2×3 mesh NoC.

The *all-to-all* traffic consists on all PEs sending data to its neighbor, simultaneously; and afterwards generating traffic for the next PE, in sequence. This aims at avoiding that all PEs send traffic to the same destination, which would generate a bottleneck in the local input of the target PE. We implemented a traffic variation that spreads the communication and denominate this type of traffic *all-to-all complement*. Every PE starts sending packets to each complement, and sequentially increments the target address. The increment rule follows the XYZ sequence; i.e., considering a 3×2×3 NoC, the PE 000 starts sending packet to PE 212, than it performs the following target address sequence {100, 200, 010, 110, 210, 001, 101, …, 012, 112, 212, 100, …}. This pattern continues in execution until packets are available to be transmitted; i.e. when application data needs to be transmitted.

We have also implemented the *dataflow* traffic pattern for evaluating distributed arbitration. It creates parallel flows of traffic through as many ports as possible, in each routing unit. Figure 9 illustrates this traffic, where west PEs (i.e. connected to WEST port) send data to east PEs and vice-versa; bottom PEs send data to the top PEs and vice-versa; and finely north PEs send data to south PEs and vice-versa.

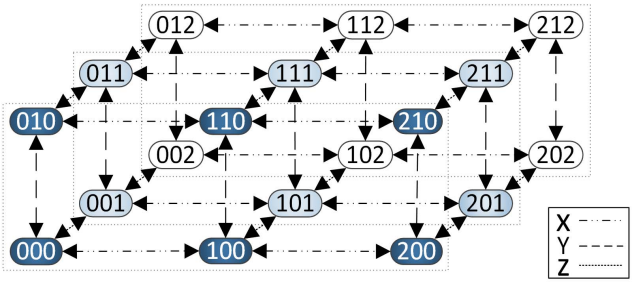


Figure 9 – Example of dataflow traffic pattern in a 3×2×3 mesh NoC.

The dataflow traffic occurring simultaneously in both directions, with small sized packets (e.g. 3-flits), floods the NoC

with switching requests, thus increasing routers demand, which enables to evaluate the router behavior when facing numerous parallel switching requests.

V. Experimental Results

Our experiments focus on network and application latencies, as well as the silicon area overhead of the proposed routers.

A. Latency Results

Latency is highly dependent on PE/task mapping. Our chosen deterministic traffic patterns simulate contention situations with high NoC loads, as well as low traffic scenarios, depending on the adopted injection rates. Considering the experimental setup discussed in Section IV, we compare both network and application latencies for every NoC configuration.

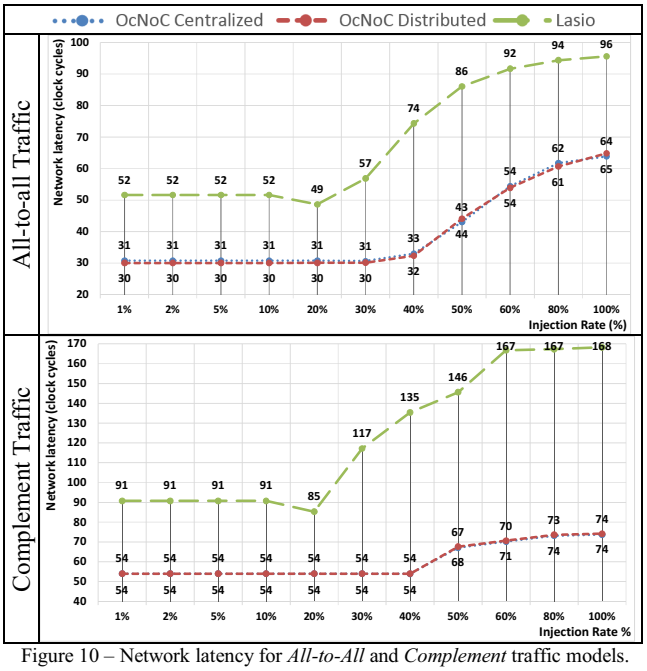


Figure 10 – Network latency for *All-to-All* and *Complement* traffic models.

In this paper, we define the network latency as the transmission delay of a packet from the local input port of source router to the local output port of the destination router, which can be influenced by other packets concurring for NoC resources. Application latency expresses the time between packet creation and packet consumption by the destination PE. We consider application latency as the most important metric for evaluating NoC communication performance, since it corresponds to the time difference of the planned injection of a packet to its delivery at the destination PE [13].

Figure 10 illustrates the measured network latency in each NoC configuration for all-to-all and complement traffic patterns. OcNoC with centralized arbitration shows an average reduction in network latency of 41% when compared with Lasio for all-to-all traffic, with a reduction of 42% when using distributed arbitration. Using complement traffic, OcNoC with centralized and distributed arbitration achieved an average reduction of 48% in network latency. Considering the latency reduction, the NoC more data traffic was necessary to achieve its saturation point. In fact, saturation occurs at about 20% of injection rate for Lasio, and at around of 40% for the OcNoC, independent on the arbitration model.

The saturation point in the NoC communication determines when application latency starts to increase due to packet contention. This behavior is observed when analyzing application latency in Figure 11 compared with the network latency in Figure 10. For lower injection rates, where data contention does not occur, application and network latency measurements present the same results. However, as more traffic is injected in the NoC, application latency increases.

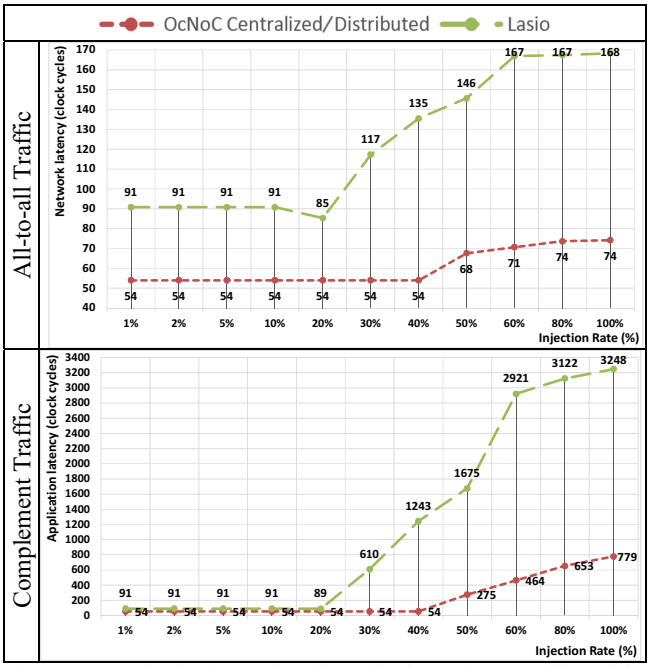


Figure 11 – Application latency for *All-to-All* and *Complement* traffic models.

As expected, from the network latency results previously discussed, application latency in OcNoC is constantly inferior to Lasio, demonstrating an average reduction of 57% for all-to-all traffic with either centralized or distributed arbitration, and up to 64% reduction in complement traffic. While OcNoC's single cycle routing shows considerable gains compared to a multistage routing implementation, our proposed distributed arbitration mechanism does not show significant gains versus centralized arbitration. It is our understanding that the adopted traffic models do not benefit from the added switching parallelism in distributed arbitration, as traffic tends to concentrate on specific channels, creating more contention as injection increases.

Using dataflow traffic, we measured network and application latency for OcNoC with both arbitration mechanisms, as shown in Figure 12. For the reported traffic model, distributed arbitration shows 24% network latency reduction in the worst case with 100% packet injection rate and with application latency decreasing by 11% when compared to centralized arbitration. On average, network and application latencies are reduced by 11% and 8%, respectively.

These results indicate that distributed arbitration benefits scenarios where routers have to handle multiple switching requests in

parallel without contention situations.

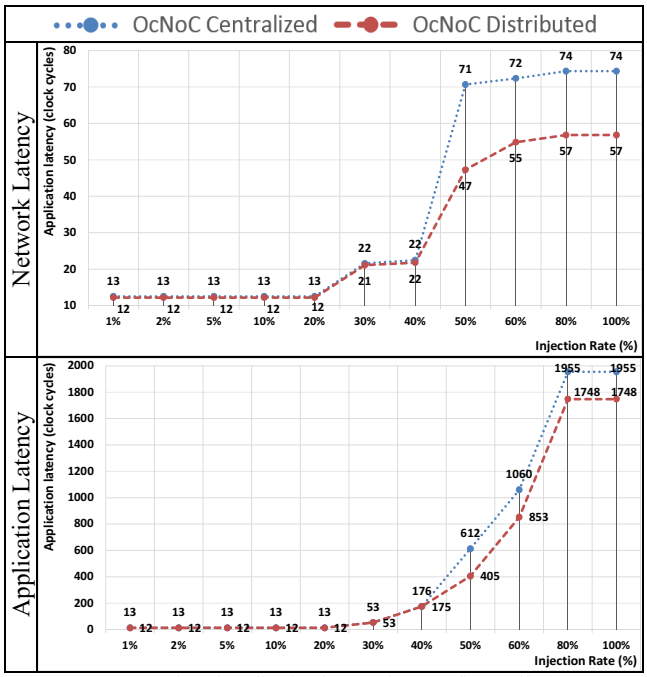


Figure 12 – Network and application latency for *Dataflow* traffic in OcNoC.

B. *Synthesis Results*

In order to evaluate area consumption, Lasio and OcNoC routers were synthetized using 65 nm STMicroelectronics CMOS technology, assuming the NoC configurations presented in Section IV. Syntheses were performed with Cadence RTL Compiler, employing a general-purpose standard cell library provided by the foundry.

Table 2 illustrates the results calculated from routers of Lasio and OcNoC with centralized and distributed arbitration. They occupied 0.072 mm², 0.07 mm² and 0.073 mm², respectively, showing that OcNoC with centralized arbitration decreases area overhead by 3%, while distributed arbitration increases area consumption by less than 1%.

Syntheses results also evaluate the combinational circuit critical path, which is an important metric for determining NoC's maximum operation frequency. Despite the increased logic circuit in the OcNoC, the critical path is measured as 1.162 ns and 1.884 ns for centralized and distributed arbitration, respectively, while Lasio's delay is 1.156 ns.

TABLE 2 – ROUTER SYNTHESIS RESULTS.

NoC	Router area (mm^2)	Critical path (ns)
Lasio	0.072	1.156
OcNoC Centralized	0.07	1.162
OcNoC Distributed	0.073	1.884

Table 2 shows that router area is slightly reduced for centralized OcNoCs and suffers very little increase concerning distributed systems. It also depicts that the critical path increases by less than 1% for centralized OcNoC routers, whereas distributed arbitration increases by 62%.

VI. CONCLUSION

Design of 3D NoCs relies on communication performance improvements, which involve changes to the router microarchitecture depending on other NoC resources and parameters. OcNoC provides one-cycle router architecture that reduces network and application latency for either low or high traffic situations, increasing the overall NoC efficiency.

Despite the larger combinational circuitry involved in our single cycle implementation, area is maintained approximately the same when compared to Lasio's multistage architecture, also roughly maintaining the same circuit delay when considering OcNoC with centralized arbitration.

Still, further explorations are necessary to evaluate the advantages of our distributed arbitration model, which benefits from highly parallel, non-competing routing traffic situations.

ACKNOWLEDGMENT

This work is partially funded by FAPERGS (Docfix SPI n.2843-25.51/12-3 and PqG 12/1777-4) and CNPq-Brasil (process number 132778/2014-9).

REFERENCES

[1] G. Luo-Feng, D. Gao-Ming, Z. Duo-Li, G. Ming-Lun, H. Ning, S. Yu-Kun. **Design and Performance Evaluation of a 2D-Mesh Network on Chip Prototype Using FPGA**. *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 1264-1267, 2008.

[2] T. Ye, L. Benini, G. De Micheli. **Packetization and Routing Analysis of On-Chip Multiprocessor Networks**. *Journal of Systems Architecture*. vol. 50, n. 2-3, pp. 81-104, Feb. 2004.

[3] P. Garrou, C. Bower, P. Ramm. **Handbook of 3D Integration**. *New York: Wiley-VCH*, Oct. 2012. 799p.

[4] T. Bjerregaard, S. Mahadevan. **A Survey of Research and Practices of Network-on-Chip**. *ACM Computing Surveys*, vol. 38, n. 1, pp. 1-51, Jun. 2006.

[5] Y. Ghidini, T. Webber, E. Moreno, F. Grando, R. Fagundes, C. Marcon. **Buffer Depth and Traffic Influence on 3D NoCs Performance**. *IEEE International Symposium on Rapid System Prototyping (RSP)*, pp. 9-15, 2012.

[6] C. Gomez, M. Gomez, P. Lopez, J. Duato. **An Efficient Switching Technique for NoCs with Reduced Buffer Requirements**. *IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 713-720, 2008.

[7] J. Kim. **Low-Cost Router Microarchitecture for On-Chip Networks**. *IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pp. 255-266, 2009.

[8] S. Nguyen, S. Oyanagi. **A Low Cost Single-Cycle Router Based on Virtual Output Queuing for On-chip Networks**. *Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD)*, pp. 60-67, 2010.

[9] M. Lai, L. Gao, S. Ma, X. Nong, Z. Wang. **A practical Low-Latency Router Architecture with Wing channel for On-Chip Network**. *Microprocessors and Microsystems*, vol. 35, n. 2, pp. 98-109, 2011.

[10] Y. Chen, Z. Lu, L. Xie, J. Li, M. Zhang. **A Single-Cycle Output Buffered Router with Layered Switching for Networks-on-Chips**. *Computers & Electrical Engineering*, vol. 38, n. 4, pp. 906-916, Jul. 2012.

[11] S. Hassan, S. Yalamanchili. **Centralized Buffer Router: A Low Latency, Low Power Router for High Radix NoCs**. *IEEE/ACM International Symposium on Networks on Chip (NoCS)*, pp. 1-8, 2013.

[12] G. Jonna, J. Jose, R. Radhakrishnan, M. Mutyam. **Minimally Buffered Single-Cycle Deflection Router**. *Design, Automation and Test in Europe Conference and Exhibition (DATE)*, pp. 1-4, 2014.

[13] E. Moreno, C. Marcon, N. Calazans, F. Moraes. **Arbitration and Routing Impact on NoC Design**. *IEEE International Symposium on Rapid Systems Prototyping (RSP)*, pp. 193-198, 2011.