

# Cluster-based Architecture Relying on Optical Integrated Networks with the Provision Of a Low-latency Arbiter

Felipe Göhring de Magalhães<sup>†\*</sup>, Fabiano Hessel\*,  
Odile Liboiron-Ladouceur<sup>†</sup> and Gabriela Nicolescu<sup>†</sup>

<sup>†</sup>Ecole Polytechnique de Montreal, Canada - <sup>\*</sup>PPGCC/PUCRS, Porto Alegre, Brazil - <sup>‡</sup>McGill University, Canada

Contact: felipe.magalhaes@acad.pucrs.br

**Abstract**—State-of-art Multiprocessor Systems-on-Chip (MP-SoC) struggle to respect the increasing requirements of high performance interconnects for high throughput communications. Optical integrated Networks (OIN) represent currently one of the most promising paradigms for the design of such next generation MPSoC. They provide increased bandwidth and better reaction to electromagnetic noise while decreasing latency and power consumption. In this paper we propose a new cluster-based architecture, the Hybrid Torus MPSoC (HTM). We propose also a low-latency arbiter allowing to exploit the full potential of the HTM architecture. Our experiments, based on FPGA prototyping and simulation, show the efficiency of the proposed architecture.

**Index Terms**—Optical Integrated Networks; Low-Latency Arbiter; Cluster-based Architectures

## I. INTRODUCTION

Modern systems have their implementation based on multiple integrated processing elements, running at a lower clock frequency, due to energy consumption constraints. Such integrated system is called Multiprocessor System-on-Chip (MPSoC). Since the introduction of MPSoCs, one of the design's main concerns lies in how the communication between internal components is performed. Electrical networks-on-chip (eNoCs) provide good communication performance [37] while maintaining an improved energy efficiency and a high re-usability level [6]. However, as the number of possible integrated cores on a single chip continues to increase, metallic interconnects in eNoCs will become a bottleneck leading the ITRS (International Technology Roadmap for Semiconductors) [1] to point out the need for a new technology to overcome such restrictions.

In this design context, on-chip optical interconnects and 3D die stacking are currently considered to be the two most promising new paradigms. Optical Integrated Networks (OINs) have already been proven to be feasible for inter-chip communications [4][7], and previous work presented photonic architectures with low power consumption, low insertion loss (7.9 dB for an 8×8 structure) and a power penalty of less than 1 dB [25]. These works bring forward OINs as attractive candidates for high demanding architectures.

Several works presented cluster-based architectures that make usage of different communication infra-structures in order to extract the best of each, such as [39][12]. This type of architectures alleviates the clock skew [27] as the components are grouped, reducing the complexity to design the clock-tree. For the same reason, the employment of Global

Asynchronous Local Synchronous (GALS) design technique [9] is facilitated when using cluster-based architectures, as each cluster might execute under its own clock domain. As cluster-based architectures rely on more than one communication infrastructure in the same system, fast interfaces and control techniques are needed. This challenge was previously addressed for the electrical communication structures, but OIN-based architecture still lack better solutions.

This work presents two main contributions:

- 1) A cluster-based architecture, the Hybrid Torus MPSoC (HTM). The HTM architecture uses electrical components to perform in-cluster communications and optical components for the intra-cluster communications.
- 2) A low-latency arbiter used to control efficiently the OIN. This controller allows the exploitation of the full potential of the HTM architecture.

Results show the efficiency of the introduced architecture as well as the low-latency impact of its arbiter. Different traffic patterns and traffic injection configurations are used to evaluate the performance.

The remaining of the paper is organized as it follows. The next section brings the state-of-art revision, positioning our approach with existing works. In Section III, the proposed cluster-based architecture is presented, followed by Section IV in which the arbiter is illustrated. In Section V, the obtained results are presented and finally, Section VI concludes this work.

## II. RELATED WORK

### A. Cluster-Based Architectures

Two main approaches are currently employed for the design of embedded cluster-based architectures: (1) defining architectures using different communication approaches, such as buses, shared memory and NoCs and (2) architectures based on hardware modifications, like embedded virtualization.

The most common approach is the usage of buses for inter-cluster communications and NoCs for intra-cluster communications. In [39] the authors propose a cluster-based system formed by ARM processors grouped in 'n' clusters of variable size which communicate through an AMBA-AHB bus [2]. The communication between clusters is performed by a NoC with a cluster on each router. An architecture using a NoC as the communication infrastructure between clusters and a simple bus to communicate internal elements on each cluster

is presented in [10]. In this model, each node is composed by IPs that may be hardware or soft cores and two modules to communicate with the NoC router.

Another class of architectures are the one based on shared memories for local communications. The architecture presented in [15] is composed of 17 processors organized into four clusters of four processors each, plus a central processor that controls the whole system. In [38] the authors present a model of dynamic clusters where the message exchange is performed using local memories. For both works, the internal modules of each cluster use a memory and the clusters are connected through a point-to-point connection.

Some previous works relied on modifications in the hardware to keep only one communication infrastructure. The authors in [33] presented a cluster-based system using a single NoC with modified routers. Unlike all previous works, the inner IPs of each cluster are directly connected to the NoC router. To do so, the NoC routers were changed to include three more local ports.

The work presented in [3] introduces a different approach for embedded cluster-based system. Virtualization techniques are used to expand the MPSoC capability, running 'n' virtual processors on each physical unit. This solution uses MIPS processors [30] and a central NoC to perform the communication between physical cores.

The works presented in [20], [23] present cluster-based architectures relying on OINs. The used approach and obtained results are very similar for both cases, where cluster composed for four IPs are connected using NoC and the clusters are connect through a  $10 \times 10$  optical switch fabric. The work [36] uses hybrid routers (optical-electrical) in the system, where the clusters are connected using an OIN organized in a butterfly fashion. This work also uses  $10 \times 10$  optical switch fabrics.

Finally, the works introduced in [32], [29] use a ring-based optical network to connect clusters, where wavelength division is employed to have an non-blocking OIN. Their approach is based on two layers: the electrical layer has the clusters and the optical layer connects the clusters using MR-based optical switches.

As it is possible to verify, systems relying solely on electrical interconnects are majority. Also, most works have pre-defined in-cluster organizations, decreasing the flexibility to use them.

Comparing with previous works, the originality of the proposed HTM architecture lies in the fact that a high-bandwidth OIN is used to connect clusters. Also, the clusters' constitution is not hard-limited or defined, which expands the design space exploration. Still, besides the internal cluster organization, the difference for the architectures that also utilize OINs lies in the employed OIN. This work relies in well established torus architectures, while the previous works use in-house OINs. Moreover, the ring-based OIN system has a constraint in the number of possible wavelengths used, thus jeopardizing its usage.

## B. OIN Controlling Solutions

Most of the state-of-art controllers are topology - or architecture specific, thus optimizing at most the performances of a specific network.

In [31], the control unit used is based on the circuit-switched algorithm. A dimension order algorithm is applied on an electrical-layer and closes the paths for the optical layer. A similar technique is used by [22], where the controlling scheme is also based on circuit-switching. The latency was calculated to be around 3.5ns on a  $8 \times 8$  network, where resonators and peripherals run at 5 GHz. An electrical-optical mixed approach is presented by [21], where Optical switches are in charge of transmitting data on a circuit-switching fashioned way, while electrical switches are in charge of closing the path by using package-switching techniques.

The control unit in [16] is based on wavelength-division multiplexing (WDM). Each I/O is assigned with one specific wavelength and might communicate at any time, without arbitration. Authors of [8] introduce a routing technique based on wavelength selection integrated with spatial routing. Circuit-switching technique is used along with WDM and each router is composed by a junction of a receiver bank and a modulator bank.

An asynchronous and variable-length packet switching is presented in [17]. Every IP is attributed with one exclusive label, which corresponds with each output fiber. While the message travels over the optical path, when it comes across a new network node, the message gets delayed while its label is computed by the electrical node. Further, a multi-cast scheduling control solution is proposed in [34], focusing input-queued switches based on the Weight Based Arbiter (WBA) and Time-division Multiplexing (TDM). The technique used by the authors is based on time sharing and aged-based weight calculations.

Finally, authors in [40] presented a controlling solution for contention handling based on optical-buffering, by introducing a three stages buffering method. This method uses electronically controlled wavelength routing switches in combination with optical delay lines to temporarily store data [11], [24], [35].

A good number of works make use of circuit-switching techniques, thus adding a control-latency that could shatter completely the benefits of using an OIN-based architectures. Also, most solutions are deeply attached to the controlled topology, which leverages their employment to different scenarios.

## III. THE HYBRID TORUS MPSOC

The Hybrid Torus MPSoC is designed in three different layers: the electrical communication layer, the interface layer and the optical layer. Its organization might be deployed in regular 2.5D fabrication technologies, but it is designed for the future 3D integration processes. Also, each cluster can operate in a different frequency, in a Global Asynchronous Local Synchronous (GALS) fashion.

The *electrical layer* is composed by generic IPs and communication infrastructures (networks-on-chip). Even though the HTM is not restricted to one specif NoC router, in this work, we relied on the HERMES router [14] for implementation and validation purposes. The HERMES is deployed as a mesh and is composed by routers, buffers and controllers of routers information. The router overview is presented in Figure 1. Still, the internal queue scheduling uses a priority round robin algorithm. The packet routing algorithm is the XY [28] and the packet flow protocol is credit-based. Each cluster is composed by 'x' IPs interconnected by a  $n \times m$  NoC.

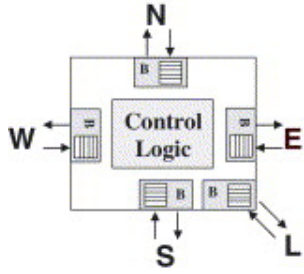


Fig. 1. HERMES router internal organization [14].

The *interface layer*, namely the *Cluster Interface (CI)*, performs the communication between each cluster and the optical layer. The CI consists of two circular queues that temporary store the data traveling between layers and a serializer/deserializer module. This module runs in parallel with the IPs executing in each cluster and in order to exchange a message between clusters, the message should pass through the CI first. On the receiver node, the CI module forwards the message to the destination IP unit.

In order to make all clusters execute independently from each other motivated the usage of buffers to temporarily store the messages. The main idea is to make the message exchange overhead smaller as possible in the application level, working in a pipeline fashioned way. Thus, when an IP is communicating with another IP that is not in the same cluster, it sends the message to the CI module and after continues its regular execution, while the CI module performs the rest of the message delivering. Figure 2 presents the fifo employed in the CI, where it is possible to see the circular infrastructure used.



Fig. 2. Cluster interface circular FIFO overview.

The optical layer is responsible for exchanging messages between each cluster and is organized in a torus topology. Its design is based on a  $5 \times 5$ , strictly non-blocking optical router [19]. Figure 3 illustrates the router internal organization where it is possible to see the 16 micro-ring resonators (MRs), six

waveguides, and two waveguide terminators. The MRs in the switching fabric are identical, and have the same on-state and off-state resonance wavelengths.

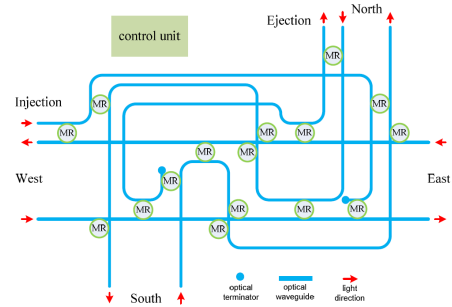


Fig. 3. Optical router organization [19]

Each cluster is composed by a variable number of IPs and each cluster can be configured with one frequency. Each cluster is connect to one optical router through the CI. Figure 4 presents the schematic overview of the HTM, where both optical and electrical layers are illustrated. In the Figure, each cluster is highlighted with a dotted box and is composed by 25 routers, in a  $5 \times 5$  NoC mesh-topology. Each electrical router in the electrical network is represented by a small circle. Still, the optical network is illustrated, where each optical router is represented as a big circle, with the CR inscriptions inside. For the sake of better viewing the CI was omitted in the Figure.

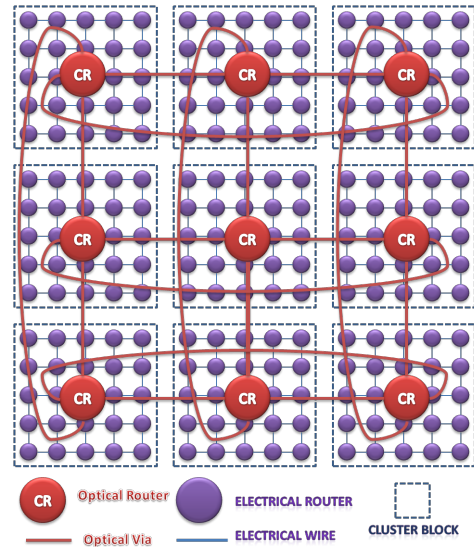


Fig. 4. HTM topology schematic overview.

#### IV. LOW LATENCY ARBITER

The performance and efficiency of OIN-based architectures can be constrained by their controllers. Long setup time of circuit-switching techniques make them not practical and, at the same time, centralized controllers have been successfully demonstrated [13], thus this is the model we are going to use.

That being said, following the blocks that compose the arbiter are presented:

- **conflicts resolution block (CRB)**: this block is responsible for detecting destination conflicts and solving them by the usage of a given algorithm;
- **memory (LUT)**: is used to store static data accessed by the controller during run-time. This memory is used mainly to reduce computation time, thus reducing control latency, and;
- **dynamic setup (DSB)**: block responsible for on-line calculations, like path attribution and memory addresses reading, by the usage of a real-time calculation (RTC) unit.

#### A. Path Analyzer and LUT Creation (PALC)

The Path Analyzer and LUT Creation (PALC) block is responsible for evaluating the path diversity of network topologies by analytically analyzing them and generate the memory-arrays to be used as addressing lists. In order to create the table, the Dijkstra algorithm [26] is used to compute the shortest possible routes. Every single possible communication scenario is evaluated in this stage and all of them are stored as look-up tables.

#### B. Conflict Resolution Block (CRB)

The CRB is a hardware block responsible for detecting conflicts (or contention) in the targeted IPs. A conflict is defined as any situation in which two or more source IPs are targeting the same destination IP, simultaneously.

The CRB works in two steps: first, it analyzes all requests, looking for a conflict, and; second, if a conflict is found, a Round-Robin (RR) algorithm is applied to define which IP will have its accessed granted. The matrix method works by checking all matrix's columns and, for the cases where more than one output is marked as one, it sets a conflict. Following, two matrices are presented : in the matrix to the left **A** is targeting **C**, **B** targets **A**, **C** targets **D** and **D** targets **B**, hence no conflicts. in the matrix to the right, **B** targets **A** but **D** is also targeting **A**, so a conflicting situation is found.

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

After, all columns  $j$  of the matrix are checked to find conflicting points, such as:

$$\forall j \in \mathcal{M}, \quad \neg XOR(j) \wedge OR(j) \implies conflict(j) = 1.$$

Furthermore, the request computation runs parallelized, in which each possible request port is considered as one running process. It made possible for the arbiter to receive requests, solve conflicts and grant access to the network with a low latency.

#### C. Dynamic Setup Block (DSB)

The LUT is created based on the possible paths a message may take on the network. It stores the route each message should follow through the network in order to reach its

destination. As it would be very costly to store every single combination of requests and targets, the Dynamic Setup Block (DSB) is used. The DSB realizes real-time calculation of paths configuration based on a minimized LUT version, the LITE-LUT. The LITE-LUT stores only portions of the network paths, like the configuration of one switch, so the LUT usage does not turn into an overhead.

## V. RESULTS

This section presents the results obtained using the arbiter and its application in the HTM architecture. Firstly, only the arbiter was simulated and also FPGA prototyped. The FPGA board was integrated with fabricated switches in order for the arbiter to be validated in realistic scenarios. Later, its latency time was compared against state-of-art works in order to analyze their impacts on the system execution. Finally, it was integrated with the HTM architecture and the performance evaluated.

#### A. Arbiter Validation and Comparison

In order to validate and evaluate the arbiter, FPGA prototyping was performed where the FPGA board was integrated with fabricated MZI-switches<sup>1</sup> [25]. Also, it was validated through VHDL-based simulations, where we used values extracted from fabricated devices. The arbiter was tested under the injection of different traffic patterns, like all-to-all, all-to-one and compliment. The design was synthesized using the proposed flow for the STMicroelectronics 65nm technology process.

The synthesis process was used in order to verify the minimum time period possible. To do so, different network sizes' arbiters were configured: 8 inputs, 16 inputs, and 32 inputs. The average minimum delay obtained for this step was  $\approx 1.4ns$ .

Figure 5 presents the simulation waveform of a  $64 \times 64$  topology. In the presented scenario, the input ports are configured to target output ports using a compliment pattern, except for the input port 1 that is targeting output port 1<sup>2</sup>. This configures a conflict, as two ports target the same output. In the Figure, it is possible to see that it takes one clock cycle for a request ( $rx$ ) to be acknowledged ( $ack$ ). Still, the conflicted port waits for the end of previous communication ( $tail$ ), which leads to a no conflict situation, and then has its request granted as well.

By having the latency measured, the real impact of the controller latency on a system was verified and compared with state of art work. For a fair comparison, the well known  $8 \times 8$  Beneš [5] topology was used. Also, taking as base a fabricated optical switch, the latency for each optical bit to pass through the network was rounded to 200 ps. The comparison was

<sup>1</sup>Mach-Zehnder Interferometer (MZI) is a device used to control the amplitude of an optical wave by diving it in two, applying a given delay and then merging the two beams of light into one [18].

<sup>2</sup>the following mapping is performed, illustrated in the Figure by the signal DEST:  $1 \rightarrow 1, 2 \rightarrow 63, 3 \rightarrow 62 \dots 62 \rightarrow 3, 63 \rightarrow 2, 64 \rightarrow 1$

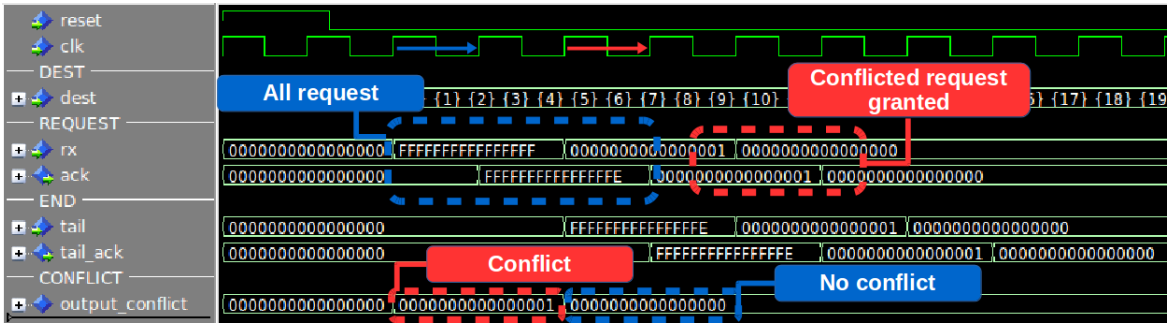


Fig. 5. Arbiter simulation waveform.

performed by analyzing the total time it takes for a message to be arbitrated and pass through the network, such as:

$$TotalTime = CL + Nob * TD, \quad (1)$$

where CL stands for control latency, Nob stands for number of bits transmitted and TD stands for transmission delay. Still, four different message sizes (128 B, 256 B, 512 B, 1 Kb) were used.

Figure 6 presents the latency comparison with four state of the art solutions [22], [21], [17], [34]. The Figure shows that the arbiter latency is comparable to the fastest presented ones. However, it still contains differences that put them apart from each other. The solution presented in [22] claims to use an operation frequency of 5 GHz, which is not realistic, so much that all validations were under simulations only. Further, the provided solution in [17] was validated through FPGA prototyping, with similar latency to LUCC. Nevertheless, its usage imposes modifications on the application network layer, which is not always possible, thus reducing its applicability. Finally, the approach used in [34] uses the same time division technique as this work, and obtained fairly similar results. The solution is suited for a specific topology, jeopardizing its usage for other cases.

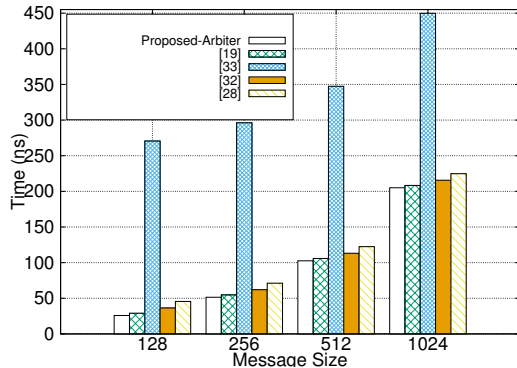


Fig. 6. Latency comparison.

### B. HTM Validation and Comparison

The Hybrid TORUS MPSoC was validated through simulation and different traffic patterns were adopted. In order

to evaluate the HTM performance, different cluster configurations were used, presented on Table I. In the Table, it is possible to see four columns: *Total I/Os* defines the total number of in/out ports in the network; *OIN Nodes* shows the number of optical routers, which is equal to the number of clusters in the system; *Cluster Nodes* holds the number of inner nodes for each cluster, and; *NoC Size* determines the intra-cluster NoC size for each cluster.

TABLE I  
HTM SIZES

Total I/Os	OIN Nodes	Cluster Nodes	NoC Size
36	4	9	3x3
100	4	25	5x5
196	9	49	7x7
324	9	81	9x9
576	9	144	12x12
900	9	225	15x15

The traffic injection was configured to insert data using different traffic patterns, such as all-to-all and compliment. Also, the injection rate was configured to the maximum frequency allowed by the components, in order to obtain the maximum available throughput. Still, different messages sizes were adopted. By analyzing the traffic scenarios, we extracted the average latency for messages to be delivered, where the latency is measured as the time between the moment the first flit is inserted into the sender and the last flit leaves the receiver.

Results show that the communication capability of the HTM matches state of art solutions, where the average bit latency is found to be  $\approx 0.91ns$  and the worst-case latency  $\approx 1.09ns$ . Based on the bits latency, the total throughput for each channel, and lastly for the entire network was measured.

The Figure 7 presents the obtained results for the HTM under different traffic patterns and messages sizes. The results are presented in a normalized form and are based on the measured latency and three different traffic patterns: compliment, local and non-uniform. It is possible to see that the HTM suits better for the compliment pattern, due the fact that the HTM is designed to improve the distant communications, leaving the local traffic for the eNoCs.



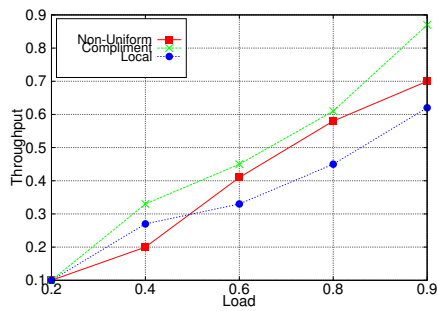


Fig. 7. Normalized throughput graph.

## VI. CONCLUSION

This work presented the Hybrid TORUS MPSoC, a cluster-based architecture for future multiprocessor systems. It takes advantage of Optical Integrated Networks' high bandwidth in order to perform long communications and the already well deployed Networks-on-chip for shorter communications. Also, an arbiter to be used in the proposed architecture was introduced. Obtained results showed a fast response time when employing the arbiter in OINs. Also, the impact on latency was tested against state-of-art works and showed that the proposed solution proved to be more efficient in most cases. Finally, the HTM's performance was presented, showing the high bandwidth obtained.

## REFERENCES

- [1] International technology roadmap for semiconductors, <http://www.itrs.net/> - last access on 12/2014.
- [2] Amba ahb reference, url: <http://alturl.com/88d98>, last access April, 2014.
- [3] A. Aguiar *et al.*, Embedded virtualization for the next generation of cluster-based mpsoCs. In *Rapid System Prototyping (RSP)*, 2011.
- [4] A.V. Rylyakov, *et al.* Silicon Photonic Switches Hybrid-Integrated With CMOS Drivers. *IEEE Journal of Solid-State Circuits*, 2012.
- [5] V. Beneš. On rearrangeable three-stage connecting networks. pages 1481–1492. *Bell Syst. Tech. J.*, 1962.
- [6] L. Benini and G. De Micheli. Powering networks on chips: energy-efficient and reliable interconnect design for socs. In *Proceedings of the 14th international symposium on Systems synthesis*, New York, NY, USA, 2001. ACM.
- [7] B.G. Lee, *et al.* Monolithic Silicon Integration of Scaled Photonic Switch Fabrics, CMOS Logic, and Device Driver Circuits. *Journal of Lightwave Technology*, 2014.
- [8] J. Chan and K. Bergman. Photonic interconnection network architectures using wavelength-selective spatial routing for chip-scale communications. *Optical Communications and Networking, IEEE/OSA Journal of*, 2012.
- [9] D. M. Chapiro. *Globally-asynchronous Locally-synchronous Systems (Performance, Reliability, Digital)*. PhD thesis, Stanford, CA, USA, 1985. AAI8506166.
- [10] D. Melpignano *et al.*. Platform 2012, a many-core computing accelerator for embedded socs: Performance evaluation of visual analytics applications. In *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, 2012.
- [11] D.K. Hunter *et al.*. Buffering in optical packet switches. *Lightwave Technology, Journal of*, 1998.
- [12] F. Magalhaes *et al.*. Embedded cluster-based architecture with high level support - presenting the hc-mpsoC. In *Rapid System Prototyping (RSP)*, 2014.
- [13] Fei Lou, *et al.* Towards a centralized controller for silicon photonic MZI-based interconnects. In *Optical Interconnects Conference - paper WD4*, 2015.
- [14] Fernando Moraes *et al.*. Hermes: an infrastructure for low area overhead packet-switching networks on chip. *Integr. VLSI J.*, 2004.
- [15] Geng Luo-feng *et al.*. Performance evaluation of cluster-based homogeneous multiprocessor system-on-chip using fpga device. In *Computer Engineering and Technology (ICCT)*, 2010.
- [16] H.A. Khouzani *et al.*. Fully contention-free optical NoC based on wavelength routing. In *Computer Architecture and Digital Systems (CADS)*, 2012.
- [17] Haijun Yang, *et al.* Design of Novel Optical Router Controller and Arbiter Capable of Asynchronous, Variable length Packet Switching. In *Photonics in Switching, PS*, 2006.
- [18] P. Hariharan. *Basics of interferometry*. Elsevier Academic Press.
- [19] Huaxi Gu *et al.*. A low-power low-cost optical router for optical networks-on-chip in multiprocessor systems-on-chip. In *ISVLSI'09 - 2009*.
- [20] Hui Li *et al.*. A hierarchical cluster-based optical network-on-chip. In *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, May 2010.
- [21] Junhui Wang, *et al.* A Highly Scalable Butterfly-Based Photonic Network-on-Chip. In *Computer and Information Technology (CIT)*, 2012.
- [22] Z. Li and T. Li. ESPN: A case for energy-star photonic on-chip network. In *Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on*, 2013.
- [23] Luying Bai *et al.*. A cluster-based reconfigurable optical network on chip design. In *Photonics and Optoelectronics (SOPO), 2012 Symposium on*, May 2012.
- [24] M. Renaud, *et al.* Transparent optical packet switching: The European ACTS KEOPS project approach. In *IEEE Lasers and Electro-Optics Society Annual Meeting*, 1999.
- [25] M.S. Hai *et al.* MZI-based non-blocking soi switches. In *Asia Communications and Photonics Conference 2014 - paper ATH3A.147*.
- [26] N. Jasika, *et al.* Dijkstra's shortest path algorithm serial and parallel execution performance analysis. In *Proceedings of the 35th International Convention MIPRO*, 2012.
- [27] P. Ramanathan *et al.*. Clock distribution in general vlsi circuits. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, May 1994.
- [28] S. Pasricha and N. Dutt. *On-Chip Communication Architectures: System on Chip Interconnect*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2008.
- [29] S. Pasricha and N. Dutt. Orb: An on-chip optical ring bus communication architecture for multi-processor systems-on-chip. In *Design Automation Conference, 2008. ASPDAC 2008. Asia and South Pacific*, pages 789–794, March 2008.
- [30] S. Rhoads. Mips plasma, url: <http://opencores.org/project>, last access July, 2014.
- [31] Ruiqiang Ji *et al.*. Five-port optical router based on microring switches for photonic networks-on-chip. *Photonics Technology Letters, IEEE*, 2013.
- [32] S. Le Beux *et al.*. Optical Ring Network-on-Chip (ORNOC): Architecture and design methodology. In *Design, Automation Test in Europe Conference Exhibition (DATE), 2011*, 2011.
- [33] M. Seifi and M. Eshghi. A clustered noc in group communication. In *TENCON*, 2008.
- [34] M. Shoaib. Selectively weighted multicast scheduling designs for input-queued switches. In *Signal Processing and Information Technology, 2007 IEEE International Symposium on*, 2007.
- [35] T. Sakamoto *et al.* Variable optical delay circuit using wavelength converters. *Electronics Letters*, 2001.
- [36] X. Tan, M. Yang, L. Zhang, X. Wang, and Y. Jiang. A hybrid optoelectronic networks-on-chip architecture. *Lightwave Technology, Journal of*, 32(5):991–998, March 2014.
- [37] Tota, S. *et al.*. A multiprocessor based packet-switch: performance analysis of the communication infrastructure. In *Signal Processing Systems Design and Implementation, 2005. IEEE Workshop on*, 2005.
- [38] M. Tudruj and L. Masko. Dynamic smp clusters with communication on the fly in soc technology applied for medium-grain parallel matrix multiplication. In *Parallel, Distributed and Network-Based Processing, 2007. PDP '07. 15th EUROMICRO International Conference on*, 2007.
- [39] Xin Jin *et al.* Fpga prototype design of the computation nodes in a cluster based mpsoC. In *Anti-Counterfeiting Security and Identification in Communication (ASID), 2010 International Conference on*, 2010.
- [40] Y. Liu, *et al.* All-optical buffering using laser neural networks. *Photonics Technology Letters, IEEE*, 2003.