BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **14**, No 1

Sofia • 2014

Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2014-0008

Towards a Design Space Exploration Methodology for System-on-Chip

A. Chariete, M. Bakhouya**, J. Gaber*, M. Wack**

** Universite de Technologie de Belfort Montbéliard Rue Thierry Mieg, 90010 Belfort Cedex France*
*** International University of Rabat ParcTechnopolis, 11 100 Sala el Jadida, Morocco*
Emails: abderrahim.chariete@utbm.fr mohamed.bakhouya@uir.ac.ma gaber@utbm.fr maxime.wack@utbm.fr

Abstract: *This paper provides an overview of a design space exploration methodology for customizing or tuning a candidate OCI architecture, given a resources budget and independent of a particular application traffic pattern. Three main approaches are introduced. The first approach allows customizing the On-Chip Interconnect by adding strategic long-rang links, while the second consists in customizing the buffer sizes at each switch according to the traffic. The third approach uses a feedback control-based mechanism for dynamic congestion avoidance. Some results are presented to shed more light on the usefulness of these approaches for System-on-Chip design.*

Keywords: *System-on-Chip, On-chip interconnect, simulation, performance evaluation, design-space exploration.*

1. Introduction

Network-on-Chip (NoC) has emerged as a solution of non-scalable shared bus schemes currently used in SoC design [2, 5]. The On-Chip Interconnect infrastructure (OCI) represents one of most important components in determining the overall performance (e.g., latency and throughput), reliability and cost (e.g., energy consumption and area overhead) of future SoCs. Furthermore, the increasing complexity of OCI infrastructures makes their design extremely challenging. OCI

topology is a very important feature in the design of NoC because the router design depends on its characteristics (e.g., diameter, average distance, clustering degree).

Many studies have shown that OCI architectures must be customized at design time in order to improve the performance of a specific application domain [1, 15]. These approaches are generally tailored for a specific application by providing a customized SoC. They deal with the selection of OCI architecture to accommodate an expected application-specific data traffic pattern. The customizing approaches can be classified based on the level, at which the customization is carried out, i.e., at application level, at communication level or at physical level, as illustrated in Fig. 1. At physical level, the on-chip interconnect configuration, bandwidth allocation, and buffer minimization are three main issues that should be addressed. At the communication level, switching, network flow control, and data routing techniques should be carefully designed. At the application level, mapping and scheduling application tasks while optimizing the cost and performance metrics constitute the main issues.

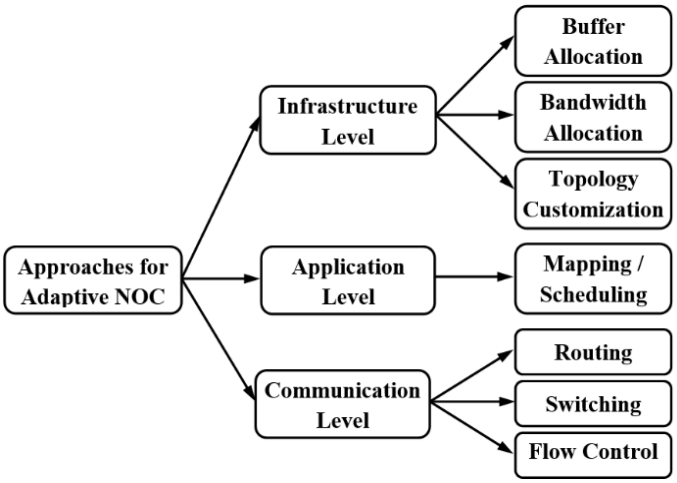

Fig. 1. Classification of customizing approaches [1]

2. Related work

Recent studies have shown that none of OCIs could provide the best performance for a wide range of applications. In other words, several SoC studies have chosen 2D mesh as underlying topology because of its regularity and low hardware complexity [2]. Other topologies (e.g., FT, BFT, Spidergon, WK) have been adapted for SoC design [3-5, 13], however, there is no universal OCI, which could support all SoC application traffic patterns.

Recently, there has been a great deal of interest in the development of analytical performance models for NoC design. The approaches proposed in literature can be classified in four main categories: deterministic approaches, probabilistic approaches, physics based approaches, and system theory based approaches. In the first category, the approaches are mainly based on graph theory

used successfully in many software and computer engineering domains. Deterministic approaches assume that the designer has thorough understanding of the pattern of communication among cores and switches. Most of the work to date using probabilistic approaches, is based on queuing theory, quantum-like approach, statistical physics and information theory [11, 17]. The fourth category uses the system theory that is successfully applied to design electronic circuits. Network Calculus is derived from system theory and has attractive features, such as the ability to capture all traffic patterns with the use of bounds, which allows the designers to capture some dynamic features of the network [10, 18].

In our recent studies, a design space exploration methodology is introduced in [6-9] for customizing or tuning a candidate OCI architecture, given a resources budget and independent of a particular application traffic pattern. This paper provides an overview of this work and highlights the main results.

The remainder of this paper is organized as follows. Section 2 describes the links insertion approach and its evaluations. In Section 3, a buffer-space allocation approach is presented. Section 4 describes the congestion avoidance approach. Conclusions and future work are given in Section 5.

3. Links insertion approach

Numerous studies have shown that in order to improve the performance and reduce the energy consumption for a specific application domain, the network architecture could be customized by inserting a number of links between routers. Several OCI-based (e.g., 2D mesh, Spidergon) architectures are recently studied and adapted for SoCs. These OCIs have different features based on different criteria [3], which are defined as follows:

- The Diameter is the largest number of hops among all shortest paths.
- The Average Distance is the average number of hops between all nodes.
- The Degree is the number of direct neighbours of a node in the OCI.
- The Bisection is the minimum number of links to be removed to separate the network into two equal portions.
- The Number of Links is the number of bidirectional links in the OCI.
- The Clustering Degree is used to specify how nodes are interconnected to each other.

These criteria could provide an initial insight on some performance metrics. For example, a small diameter and a low average distance allows fast communication between farthest nodes. A high degree allows many close neighbours perform fast communications. A higher clustering degree indicates that nodes close to each other are strongly connected. Thus, if an OCI architecture has a small diameter, a high degree, and a high clustering degree, then it could support various data traffic patterns.

In order to show the efficiency of the inserting links between the strategic nodes, a theoretical evaluation has been first conducted using MatLab. Indeed, many topologies (Ring, 2D Mesh, Torus, WK, X-Mesh and Spidergon) have been

evaluated using their topological criteria, mainly the Average Distance and the Clustering Degree criteria. It was shown that Ring topology is the best OCI when it is increased by additional strategic links. However, it was noticed that almost all topologies have a similar average distance and clustering degree, after adding about 30% of links when compared to a fully connected topology.

Simulations have been also conducted using Nirgam, a simulator dedicated for evaluating NoCs [12].It is an accurate simulator that provides substantial support to an experiment with various options available at every stage of NoC design in terms of topology, switching technique, virtual channels, buffer parameters, routing mechanism and pattern traffic [7].

Figs 2 and 3 illustrate the evaluation of the average latency and the energy consumption respectively when using Transpose traffic pattern (i.e., the node with binary value *a* (*i*, *j*) communicates with the node *a* (*j*, *i*)) and the flit-interval equals 10, 20 and 30 clock cycles on 2D Mesh NoC topology. The results depicted in these figures show 20% and 15% improvement in terms of latency and energy consumption respectively when only 12 links were added to the basic OCI [6].

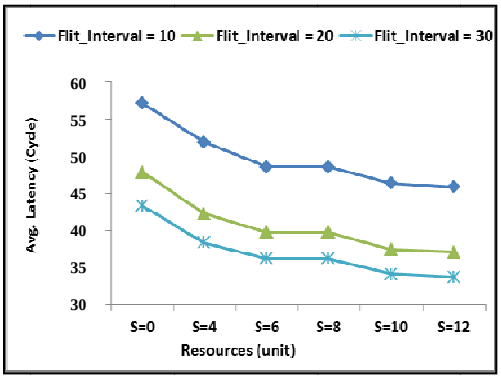

Fig. 2. Average Latency vs. resource budget

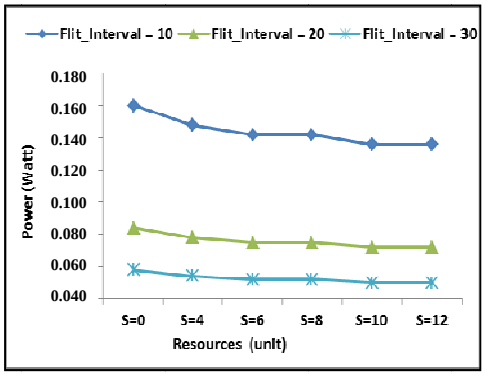

Fig. 3. Power consumption vs. resource budget

This study allows us to propose a new fractal topology, called FracNoC [7]. This topology was evaluated and compared with 2D mesh and Torus for different traffic patterns (e.g., Shuffle, Bit-Reversal, Uniform and Transpose). Fig. 3 illustrates the evaluation of the average latency and the energy consumption

respectively when using a Transpose traffic pattern. The results obtained show that FracNoC outperforms 2D mesh and Torus topologies. However, it provides similar results for other traffic patterns when compared to Torus.

Then, an thorough comparative study of topologies (2D Mesh, X-Mesh, Torus, Spidergon, WK and FracNoC) increased by adding strategic links was presented in [8]. Simulations have been conducted using several traffics, such as Transpose, Bit-Reversal, Shuffle and Uniform. The obtained results showed the efficiency of the fractal-like topologies (WK and FracNoC) as a communication fabric for SoC design.

4. Buffer space allocation approach

The work presented in this section is focused on optimizing the buffer spaces inside switches. Using buffers with a fixed size generates congestion at routers, which increases the energy consumption and has a significant influence on performance (e.g., flits drop). However, we have introduced a technique for customizing the size of buffers inside switches [9]. It is a design space exploration approach to allow customizing an on-chip interconnects architecture that matches the workload-specific application of a System-on-Chip. Indeed, only the required resources are allocated for each channel based on the traffic pattern of a target application. Fig. 4 illustrates a given traffic pattern for 2D Mesh.

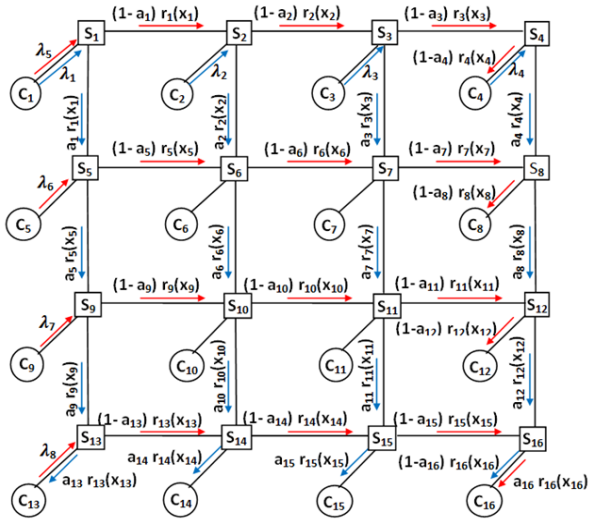

Fig. 4. A traffic pattern for 2D Mesh

The approach uses the compartmental Fluid-flow based theory [14] to model the system and then allocate the required resource for each buffer. Simulations have been conducted and some results are depicted in Fig. 5. These results showed that by analyzing and capturing the characteristics of on-chip communication traffic, the designer can select and design the on-chip interconnect routers that are optimized for a target application.

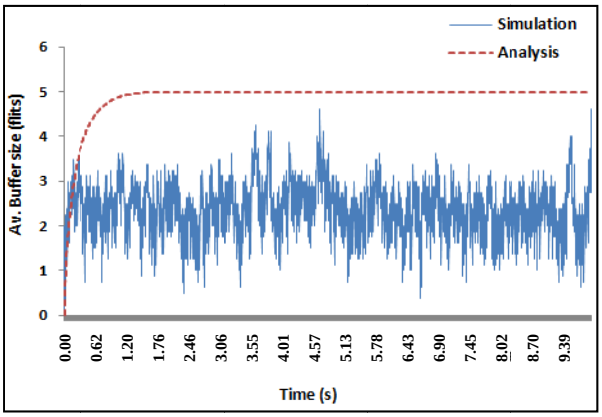

Fig. 5. The buffers size variation over time computed with simulation and analysis when the injection rate is 80 flits per s

5. Congestion avoidance approach

The approach presented in the previous section targets the application-specific SoC. However, this approach did not consider the run-time configuration of different NoC parameters, which are hard to predict at early development stages. For example, the design of NoCs that are able to handle all application requirements and to predict parameters in the early development stages require run-time approaches [1]. More precisely, since resources must be shared between multiple applications, bottlenecks may be created in some switches, and therefore could lead to poor performance.

The approach introduced in this work includes a control mechanism to allow NoC elements adjusting the data flow in order to guarantee the boundedness of the buffers size defined at design-time [10]. Fig. 6 illustrates a given traffic pattern used in our simulations on Spidergon OCI.

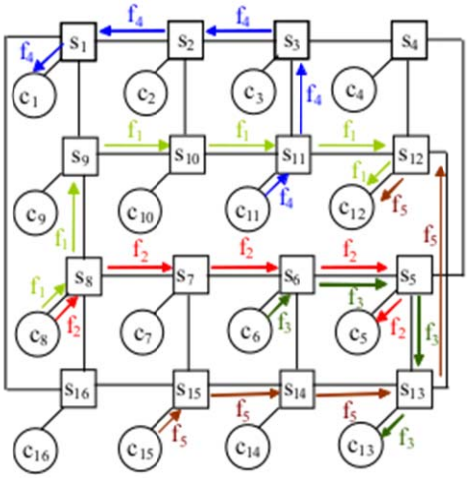

Fig. 6. A traffic pattern for Spidergon

The obtained results showed the efficiency of this control mechanism in avoiding congestion inside switches. For instance, Fig. 7 shows the buffer size occupancy with a maximum buffer size fixed to 10 flits. The results show the viability of this mechanism for congestion avoidance.

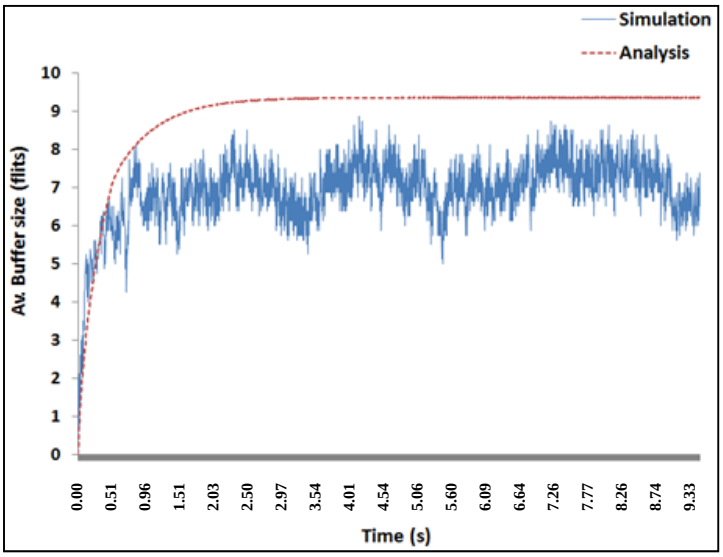

Fig. 7. The buffer size variation over time when using the feedback control mechanism with a buffer size at each switch fixed to 10 flits and injection rate – 100 flits per s

We are also investigating the use of this mechanism to develop adaptive techniques including dynamic routing data to avoid congested routers.

6. Towards fractal NoCs

Fractal architectures are receiving considerable attention in networking community. A fractal topology is a geometric structure that demonstrates similarity in properties at various scales, i.e., the structure looks similar under different magnification levels [16]. In [7], a self-similar fractal-geometry-based triangle topology, called FracNoC is proposed for SoCs. Fractal based topologies have attractive properties, such as high degree of regularity, efficient communication performance for low energy consumption, and ease of extendibility that suits NoC systems.

A FracNoC network topologies is a fractal, denoted by FracNoC(k), it can be described by the expansion level k. It can be obtained from a lozenge, for an infinite number of iterations of dividing by two the size of the lozenge and then to juxtapose the quadruplicate by their vertices to form a new lozenge (Fig. 8).

More precisely, a fractal structure reproduces itself iteratively, exhibiting invariant structural properties. In other words, the fractal describes a self-organizing mechanism.

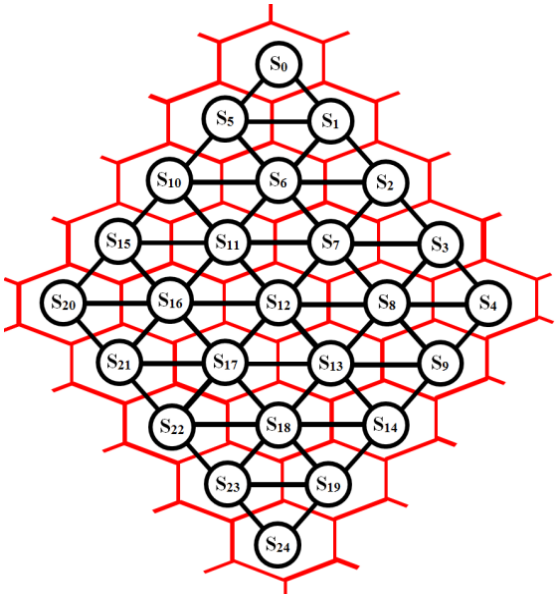

Fig. 8. A FracNoC network with $k = 4$

A FracNoC network topology is a fractal, denoted by FracNoC(k), and it can be described by an expansion level k (Fig. 9). A FracNoC network is described as follows:

- For $k = 0$ there are $N_0 = 1$ node with a maximum diameter $D_0 = 0$ and it holds $P_0 = 0$ links.
- For $k = 1$ there are $N_1 = 4$ nodes with a maximum diameter $D_1 = 2$ and it holds $P_1 = 5$ links.
- For each $k > 1$ there are $N_k = 4N_{k-1}$ nodes, with a maximum diameter $D_k = 2(D_{k-1} + 1)$ and it holds $P_k = 4P_{k-1} + 13$ links, where N_k is the total number of nodes, P_k is the total number of links, and D_k is the maximum diameter of FracNoC(k). This family of topologies, starts from a FracNoC(k) and recursively expends to any level k as illustrated in Fig. 9.

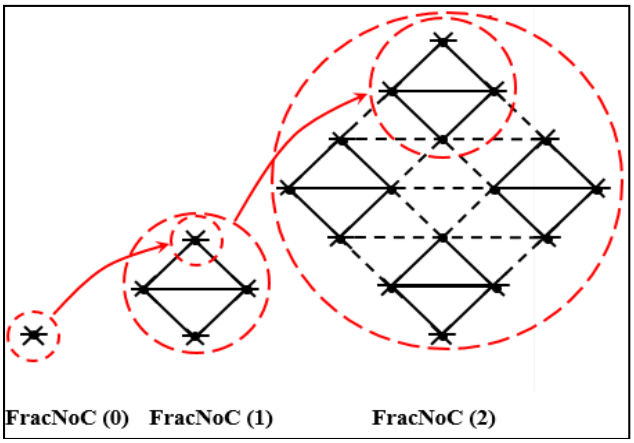

Fig. 9. A FracNoC network topology expansion (levels $k = 0$, $k = 1$, $k = 2$)

It is worth noting that FracNoC OCIs have several attractive properties, such as self-similarity, reiteration, expandability and regularity that are more suitable for NoC design. More precisely, the geometrical structure of FracNoC is factorisable.

To show the efficiency of FracNoC, simulations have been conducted using NIRGAM simulator. The following figures show comparison results with 2D Mesh and Torus concerning the throughput, the average latency, and power consumption; for a fair comparison zero flits drop is assumed. The traffic injection is varying in a clock cycle as 10, 15, 20, 25 and 30.

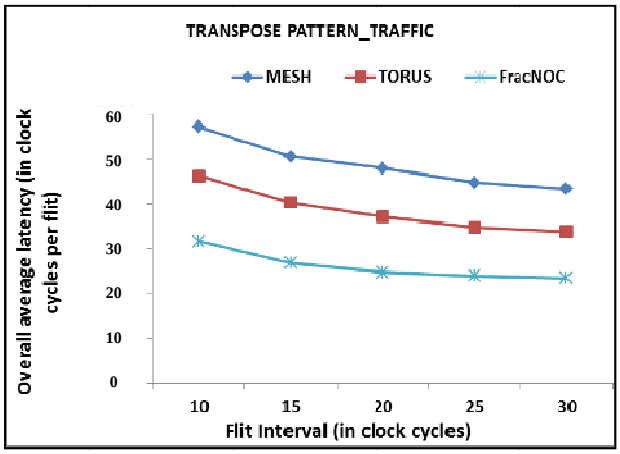

Fig. 10. Average latency comparison

Fig. 10 shows that the average latency decreases linearly in function of the traffic injection variation (inter-flits interval). FracNoC outperforms the other topologies.

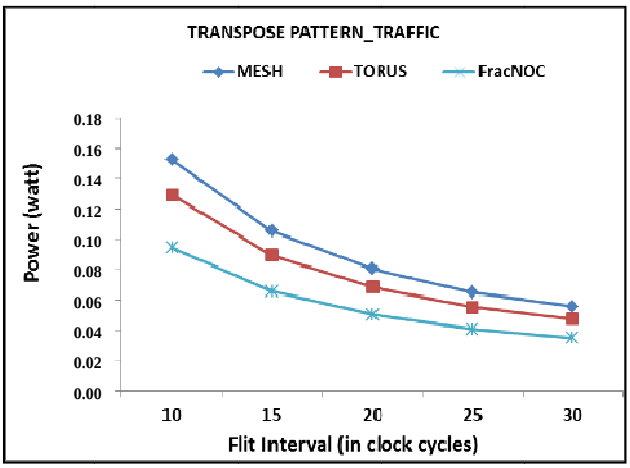

Fig. 11. Average power comparison

Fig. 11 shows the average power consumption evolution in function of the traffic workload (inter-flits interval).

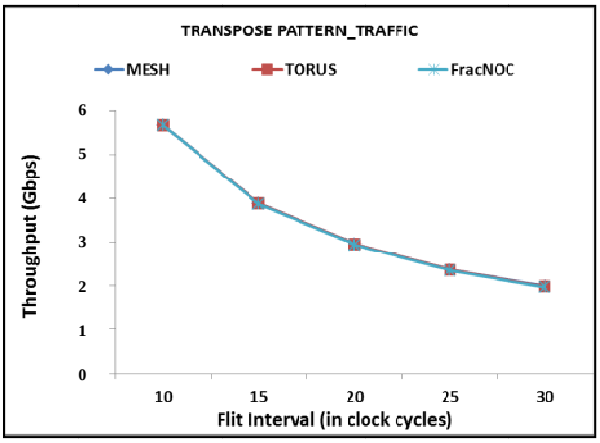

Fig. 12. Average throughput comparison

Fig. 12 shows the throughput ratios under different injection rates (inter-flits). To ensure fair comparison, flits injected by source cores to their destination cores are routed without flits losses. The objective is to make a fair comparison to put more emphasis on the influence of adding resources to a basic OCI.

7. Conclusions and future work

In this paper we have presented three approaches for customizing the on-chip interconnect that could handle emerging SoC applications. The objective of this work was firstly to build a framework combining analytical and simulation evaluation tools, NIRGAM, for OCIs customization by links insertion, customizing buffer space, or flow control for congestion avoidance. Preliminary results showed that customizing OCIs could achieve better performance while minimizing the energy consumption and silicon area overhead.

These approaches are targeted to allow designers, at the early stage of the design process, to rapidly analyze and customize a candidate on-chip interconnect with the objective in order to improve its performance. According to this evaluations study, there is no universal OCI, which could support all SoC application traffic patterns. Therefore, the run-time approaches must be included in all SoC layers, such as adaptive routing, dynamic flow control and adaptive switching techniques.

Acknowledgement: The paper is partly supported by FP7 project "Advanced Computing for Innovation" (ACOMIN), Grant Agreement 316087 of Call FP7 REGPOT-2012-2013-1.

References

1. Bakhouya, M. A Bio-Inspired Architecture for Autonomic Network-on-Chip. Autonomic Networking-on-Chip: Bio-Inspired Specification, Development, and Verification. – In: Embedded Multi-Core Systems (EMS) Book Series. Phan Cong-Vinh, Ed. Taylor and Francis/CRC. 2012, 1-20.

2. B e n i n i, L., G. D. M i c h e l i. Networks on Chips: A New Socparadigm. – IEEE Computer, Vol. **35**, 2002, No 1, 70-78.
3. C o p p o l a, M., R. L o c a t e l l i, G. M a r u c c i a, L. P i e r a l i s i, A. S c a n d u r r a. Spidergon: A Novel On-Chip Communication Network. – In: Proc. of Inter. Symposium on System-on-Chip, 2004, 250-256.
4. G u e r r i e r, P., A. G r e i n e r. A Generic Architecture for On-Chip Packet-Switched Interconnections. – DATE Proc., 2000, 250-256.
5. P a n d e, P. P., C. G r e c u, M. J o n e s, A. I v a n o v, R. S a l e h. Performance Evaluation and Design Tradeoffs for Network-on-Chip Interconnect Architectures. – IEEE Trans. on Computer, Vol. **54**, 2005, No 8, 1025-1040.
6. C h a r i e t e, A., M. B a k h o u y a, J. G a b e r, M. W a c k. An Approach for Customizing On-Chip Interconnect Architectures in SoC Design. – HPCS'12, Madrid, Spain, 2012, 288-294.
7. C h a r i e t e, A., M. B a k h o u y a, J. G a b e r, M. W a c k. FracNoC: A Fractal On-Chip Interconnect Architecture for System-on-Chip. – HPCS'13, Finland, 2013.
8. C h a r i e t e, A., M. B a k h o u y a, J. G a b e r, M. W a c k, E. C o a t a n e a, S. N i a r. A Methodology for Customizing On-Chip Interconnect Architectures. Accepted with Revision in Concurrency and Computation – Practice & Experience Journal, 2013.
9. B a k h o u y a, M., A. C h a r i e t e, J. G a b e r, M. W a c k. A Buffer-Space Allocation Approach for Application-Specific Network-on-Chip. – In: AICCSA'11, Egypte, 2011, 263-267.
10. B a k h o u y a, M., A. C h a r i e t e, J. G a b e r, M. W a c k, S. N i a r, E. C o a t a n e a. Performance Evaluation of a Flow Control Algorithm for Network-on-Chip. – In: HPCS'12, Spain, 2012, 281-287.
11. B a k h o u y a, M., S. S u b o h, J. G a b e r, T. E l-G h a z a w i, S. N i a r. Performance Evaluation and Design Tradeoffs of On-Chip Interconnect Architectures. – SIMPAT Journal, Vol. **19**, June 2011, Issue 6, 1496-1505.
12. L a v i n a, J. A Simulator for NoC Interconnect Routing and Application Modeling, 2007. **http://nirgam.ecs.soton.ac.uk/**
13. S u b o h, S., M. B a k h o u y a, J. G a b e r, T. E l-G h a z a w i. An Interconnection Architecture for Network-on-Chip Systems. – Telecom Systems, Vol. **37**, 2008, No 1-3, 137-144.
14. G u f f e n s, V., G. B a s t i n, H. M o u n i e r. Fluid Flow Network Modeling for Hop-By-Hop Feedback Control Design and Analysis. – In: Proceedings Internet-Working, 2003.
15. S u b o h, S., V. N a r a y a n a, M. B a k h o u y a, J. G a b e r, T. E l-G h a z a w i. Methodology for Adapting On-Chip Interconnect Architectures. – IET Computers & Digital Techniques, 2003, 1-9. doi: 10.1049/iet-cdt.2013.0021.
16. D e F l o r i o, V., M. B a k h o u y a, A. C o r o n a t o, G. D i M a r z o. Models and Concepts for Socio-Technical Complex Systems: Towards Fractal Social Organizations, Systems Research and Behavioral Science. – Syst. Res. Behav. Sci., 2013, 00:1-24.
17. C h a r i e t e, A., M. B a k h o u y a, J. G a b e r, M. W a c k. Towards a Design Space Exploration Methodology for Application-Specific NoC. – In: International Conference on High Performance Computing and Simulation (HPCS 2013), 224-228.
18. B a k h o u y a, M., S. S u b o h, J. G a b e r, T. E l-G h a z a w i. Analytical Modeling and Evaluation of On-Chip Interconnects Using Network Calculus. – In: 3rd ACM/IEEE International Symposium on Networks-on-Chip (NOCS'09), 74-79.