

Adapting Communication for Adaptable Processors: A Multi-Axis Reconfiguration Approach

Paulo C. Santos¹, Gabriel L. Nazar¹, Fakhar Anjam², Stephan Wong², Luigi Carro¹

¹Instituto de Informática
Universidade Federal do Rio Grande do Sul
Porto Alegre, Brazil
{pcssjunior, glnazar, carro}@inf.ufrgs.br

²Computer Engineering Laboratory
Delft University of Technology
Delft, The Netherlands
{f.anjam, j.s.s.m.wong}@tudelft.nl

Abstract— The increasing power density found in newer manufacturing technologies dictates that it is no longer possible for the whole chip to operate at full capacity the entire time. Adaptable systems must be devised to dynamically throttle their power consumption while maintaining the high performance expected by users. Furthermore, adapting processing and memory capacities leads to variable requirements of the communication infrastructure. Thus, in order to find the best solutions in the available design space, adaptability should be applied concordantly to three system axes: processing, memory and communication. In this work, we present the case for an architecture able to dynamically adapt its performance in all such layers. We focus on providing and adaptable Network-on-Chip able to dynamically meet the requirements of a reconfigurable processor.

Keywords- Multiprocessor System on Chip; Network-on-Chip; VLIW processor; Power Consumption;

I. INTRODUCTION

With the growing need for increased performance and integration of different features in a single integrated circuit, more and more Multiprocessor Systems on Chip (MPSoCs) are emerging as a technology to provide such needs. The processing elements (PEs) that make up an MPSoC can consist of different architectures, exploiting parallelism in different granularity levels, like general purpose processors, VLIW processors, DSPs, vector processors and other accelerators. For effective communication among these processors, and also with memory, Networks-on-Chip (NoCs) have been proposed, providing reusability, scalability, parallelism and high bandwidth data communication [14].

However, the search in the design space is not only about performance, but also about consuming less power and energy. The increasing power density found in the newer manufacturing technologies makes it infeasible for the entire system to operate at maximum performance the entire time. Thus, techniques able to significantly reduce the power consumption are crucial for enabling the efficient use of aggressively scaled technologies.

Some techniques for reducing power and energy involve the reduction of frequency and voltage, others save power by changing the underlying architecture, like in the case of a

VLIW processor, for example, by reducing the number of issues that can be used. The issue-width of a VLIW processor can be reconfigured at run time [1] by temporarily enabling or disabling functional units (FUs), thus reducing the power consumed by the processor. Another benefit of reducing the issue-width is to reduce the bandwidth necessary for memories storing data and instructions. As the port width of the instruction memory is directly related to the issue-width of the VLIW processors, and cache memory is a large consumer of power [3][6], this can also be reconfigured to save power and energy [4].

The communication between the different processing elements (PEs) of a NoC-based MPSoC and the routers is done, in several cases, using the memories of these PEs, as illustrated in Fig. 1. Thus, a reduction in the issue-width of a processor could reduce the bandwidth required from its memory, and consequently, also reduce the bandwidth required from the NoC routers. Note that such changes also affect significantly the power consumption of the system. And as these changes in bandwidth are caused by a reconfiguration of the processor, any NoC planned at design time for an adaptable processor will either provide below average performance (when designed for the mean case), or dissipate excessive power (when designed for the worst case). Hence, reconfiguration at processor and memory levels must be followed by reconfiguration also at the communication level, in order to allow the use of a NoC that better suits the current processor configuration.

This article presents a study in which the NoC routers can be reconfigured in search for a better power consumption at the system level, as a function of the variable demand due to the reconfiguration of the processing elements, especially in relation to the instruction cache. We exploit different means

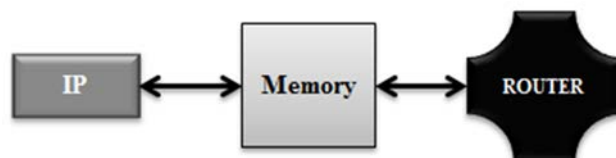


Figure 1. Noc router connected via memories

that may be used to dynamically modify the performance and power consumption of a NoC, namely scaling the operating frequency and modifying the width of the channels. These two approaches are compared in their effectiveness to reduce the overall power of the NoC.

This work is structured as follows. In section II we discuss the main related works. Section III shows how inefficient can a partially reconfigurable system be. In section IV we discuss the adaptable parameters of the proposed architecture. The experimental setup is presented in section V. Section VI analyzes the experimental results. Finally, section VII concludes the work, also pointing out relevant future work possibilities.

II. RELATED WORK

In one system, the processors are targets of several techniques for reducing power consumption and energy with minimal performance penalty. Several studies also seek other alternatives to dynamically improve the performance / power. Processors of different architectures and different purposes were proposed using techniques of reconfigurability, enabling the development of reconfigurable processing elements dynamically. Reconfigurable processors have been proposed in several works [1][16][17].

In [16], the authors propose a processor where the reconfigurable logic works as an FU. The purpose of this processor is to increase the achieved ILP (Instruction Level Parallelism) by matching the best processor configuration to the instructions that are ready to be executed. All reconfigurable processors cause variations in the demand of data at some point. In [16] the reconfiguration of the superscalar processor is an attempt to increase ILP by executing as many instructions available, and it modifies the data throughput required from the instruction memory. In [17] the authors make use of a binary translator to dynamically identify parallelizable instruction and execute them efficiently in a reconfigurable array of FUs. In [1] the ρ -VEX VLIW processor [2] has a number of issues reconfigured at run time, allowing adjustment between performance and power. Thus besides the reduction of voltage and frequency, the processor architecture can also be changed at run time for optimum use.

Another component subject of several studies is the memory, especially the cache memory. In [4] experiments are made taking into account the need to change the cache due to the variation of a reconfigurable VLIW processor [1]. These experiments show that by changing the parameters of the cache, it is possible to maintain the performance with a reduced power. The impact of different parameters in a cache memory performance and power consumption is described in [5], where a study of several caches of different processors is presented. In [6] and [7] the cache associativity is reconfigured so as to obtain a lower power consumption without a significant reduction in performance.

Besides the associativity and memory size, the size of the lines also brings changes in performance and power consumption, as shown in [8], where different line sizes show different results.

However, for a NoC-based system with maximum configurability, including processor and memory, the communication must also be capable of reconfiguration in order to follow the rest of the system. In the literature several studies can be seen, especially for the NoC routers. In [9] and [10] an implementation is shown where several parameters like routing, switching and packet size are reconfigured at run time. Thus, the NoC can be reconfigured seeking an optimal point according to the application. However, the operating frequency of the routers remains fixed. In [11] the depths of the buffers are reconfigured at runtime, showing a significant reduction in power consumption without performance penalties. Reference [12] is also concerned with the buffers of the NoC routers, where a novel reconfigurable FIFO using Johnson-encoding is used, reducing latency and power consumed by the circuit. In [13] a way to dynamically reconfigure the switching and topology of the NoC at runtime is presented. Thus traffic can be driven by reducing the critical path and latency. The solution applies primarily to FPGAs (Field Programmable Gate Arrays), bringing great challenges to enable its implementation in ASIC (Application Specific Integrated Circuit).

None of the previous works accounts for the impact of a reconfigurable processor and/or memory on the performance required from the NoC. In this work, we show that significant gains can be achieved when one couples a reconfigurable NoC that is aware of the reconfigurations performed at the processing elements. Thus, we present the case for a platform able to be configured in three axes that play significant role in the overall performance and power consumption: processing, memory and communication.

III. INEFFICIENCY OF THE PARTIALLY RECONFIGURABLE SYSTEMS

Aiming at improving the power/performance, cache memories can be reconfigured at run time following the current necessities. For example, if a processor reduces the frequency from 400MHz to 100MHz, the instruction cache memory can reduce the frequency of operation, since this obviously causes no harm to the performance.

It has been demonstrated that one can reconfigure other parameters of the cache memory, such as line size and associativity, reducing the power consumption or increasing the performance of the processing element [3][5][6][7][8]. The coupled reconfiguration of processor and memory, proposed in [4], shows interesting gains.

However, the modifications of these parameters directly influence the efficiency of the NoC as well. The NoC routers usually work in a fixed frequency and bandwidth. After the reconfiguration of the instruction cache memory, for example, the NoC can become ineffective, dissipating unnecessary power or becoming a bottleneck in the system. Some results of power consumption of the router RaSoC [15] for different data widths and frequencies are shown in Fig. 2.

Exploiting another issue, a NoC router which has a frequency of 400MHz and 128 bits of data connected to a memory with 256 bits of data running at 200MHz works with

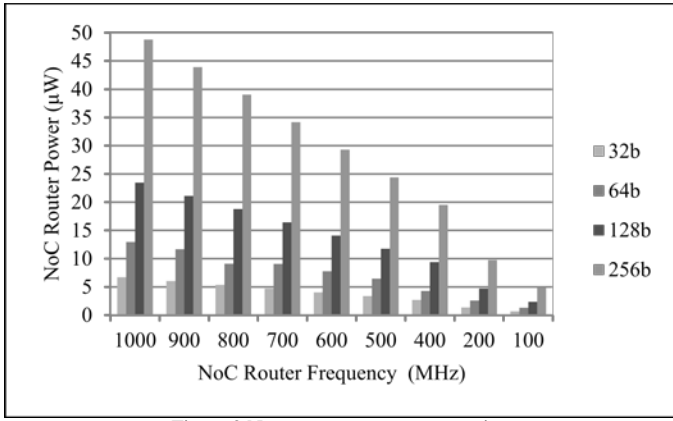


Figure 2. NoC router power consumption

maximum efficiency. But if this memory has the frequency of operation and the data width reconfigured to 100MHz and 128bit, respectively, the NoC router is consuming more power than necessary. Alternatively, if the memory data throughput is increased to 400MHz * 256 bits, the NoC will be a limiting factor for the system.

The worst cases occur when both frequency and data width are reconfigured in the PE. For such cases, the coupled effect of a wider port operating at a higher frequency will quickly cause the router to become a bottleneck of the system performance.

IV. RECONFIGURABLE PARAMETERS OF THE NOC BASED SYSTEM

In this section we present the parameters for a reconfigurable instruction cache, a reconfigurable VLIW processor and a NoC. The case study presented in this work is based on the ρ -VEX processor [1]. Note, however, that the changes observed in the bandwidth required from the communication infrastructure occur for any reconfigurable processing element.

The parameters considered herein are totally reconfigurable, allowing the exploration of a wide design space in search for the best available points.

A. ρ -VEX VLIW Processor

The VLIW processor used was the ρ -VEX, but with a contribution presented in [1]. This version of the ρ -Vex processor enables reconfiguring the VLIW processor in 2, 4 or 8 issues at runtime. This directly affects the instruction memory, since when varying the issue-width, the size of instruction also varies, requiring more or less memory bandwidth according to the modification done. In fact, regardless of the nature of the processing element, which can be a superscalar, a VLIW, a simple RISC processor, etc., reconfigurable capabilities will cause variations in data and instruction throughput requirements, which modifies the optimal memory and communication configurations.

Fig. 3 shows the base structure of the ρ -VEX and the memory and router surrounding it. Note that it allows composing larger VLIW processors with independent 2-issue nodes and that this modify the requirements of instruction

width. We assume that the processor always has exactly one load/store unit, maintaining the same requirements from the data cache, as in [4]. Note that different processor reconfiguration techniques may also lead to changes in data requirements, further increasing the relevance of the approach proposed here.

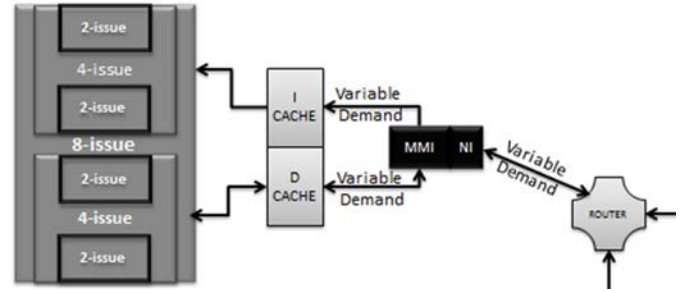


Figure 3. ρ -VEX processor, memories and router

B. Cache Memory

Our design uses cache memories that may be reconfigured at run time, allowing one to vary the port size between 64, 128 and 256 bits according to the necessity of the processing element and the performance and power required. In addition to these parameters, the operating frequency of the memory was also varied, thus simulating a complete reconfigurable PE. The size of the cache and cache lines, as well as associativity, are not discussed here, since only the throughput is taken into account in this study, and hit ratio is assumed to be constant. Other works have shown how these parameters may be dynamically reconfigured to optimize performance and power consumption. Thus, we consider a cache that modifies its throughput along with the requirements of a reconfigurable processor.

C. Network on Chip

The Network-on-chip used in this article is based on the SoCIN RaSoC router [15] with data width and frequency between 8 and 1024 bits, and 200 and 1000MHz, respectively. For this analysis we extracted estimates of power through the Cadence RTL Compiler tool using a 65nm technology. The RaSoC router allows different depths of buffers, but this parameter was kept fixed in this initial study.

V. EXPERIMENTAL SETUP

The experiment was based on a default cache 1W-16B-200 (where 1W means 1 way associativity, 16B means 16 byte line size and 200 the operating frequency in MHz). The cache memory size was fixed in 32KB as its total size

TABLE I. PARAMETERS USED IN THE EXPERIMENT

Parameters	Range
ρ -VEX issue-width	2/4/8
Cache port size (bits)	64/128/256
PE (MHz)	100/200/250/400/500
NoC Router frequency (MHz)	100/200/400/500/600/700/800/1000
NoC Router data width	8/16/32/64/128/256/512/1024

reconfigurability was not objective of this study.

The cache parameters port size and associativity are reconfigurable, allowing the variation of the maximum memory bandwidth required in each configuration. For our analysis we use the parameters shown in Table I, composing different cache settings and different configurations of NoC:

VI. ANALYSIS

In this section we study the impact of the reconfiguration of processing elements (processor and cache), with respect to performance and power in a NoC-based MPSoC, and at the same time providing reconfiguration at the NoC level. When one changes the processing power and the cache bandwidth accordingly, one has also to change the capacity of the NoC.

We observed that when we modify the bandwidth required by a PE, it is interesting that the NoC is also reconfigured, especially in terms of frequency and data width, thus achieving a system with better performance/power. Fig. 4 shows the reference node of the MPSoC for the experiment, where the PE initially runs at a frequency of 500MHz with 8-issue (256 bit instructions) and the NoC runs at 1000MHz with 128 bit data width. The MMI (Memory Management Interface) and the NI (Network Interface) are responsible for coupling these two modules that run with independent frequencies.

Based on the results presented in [4], where the configurability of p-VEX processor enables the changes in the instruction memory and data bandwidth required, three tests were made: NoC frequency variation, variation of the data width and variation both frequency and data width. The goal is to reduce power consumption in the MPSoC communication maintaining the throughput required by the PE, so that the NoC routers do not represent a bottleneck for the system.

A. Varying the frequency

The first results are related to varying the frequency of the NoC routers along with the variation of the throughput required by the PE. The variation requested by the PE can be caused by varying frequency or issue-width of the processor and/or reconfiguration of memory. Fig. 5 shows the results of power for the RaSoC router operating at various frequencies. The router varies its frequency with the PE, consuming the lowest power possible without loss of performance. This experiment fixes router data width in 128bit.

As shown in Fig.5, when the PE reduces the frequency from 500MHz to 250MHz, the NoC router reduces its frequency from 1000MHz to 500MHz while maintaining the

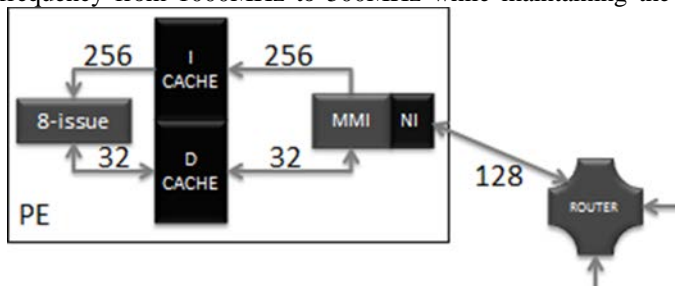


Figure 4: MPSoC reference node

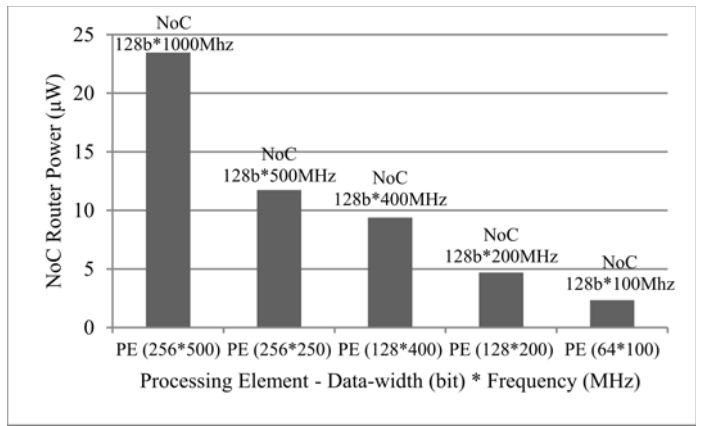


Figure 5: NoC Router Power – Frequency Variation

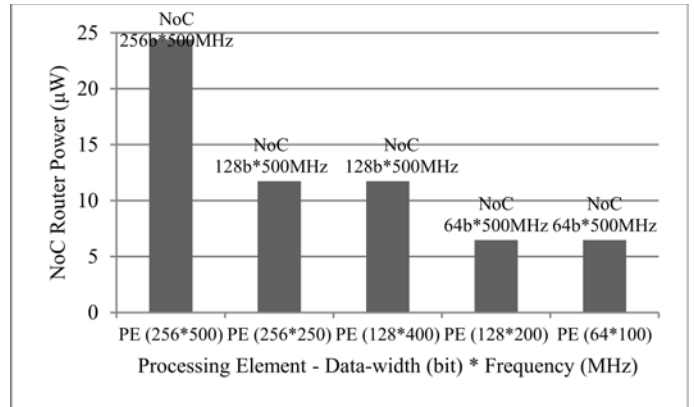


Figure 6: NoC Router Power – Data Width Variation

data throughput. This reduces the power on the router to a half. Similar situations are observed when the processor demand is further reduced, either by varying its frequency or issue width. On a non-reconfigurable NoC, the power on the router would remain the same, leading to waste of energy.

B. Varying the data width

When we vary the data width of the NoC, there is the possibility of powering off the buffers associated with the unused channel bits. Thus, relevant power reductions may also be observed with this technique. Fig. 6 shows the results of reduced power when the data width of the routers is configured. In this experiment the frequency of operation of the NoC was kept fixed at 500MHz.

As shown in Fig. 6, when the PE changes from 256 bits * 250MHz to 128 bits * 400MHz, thus reducing its needs, the NoC cannot be reconfigured precisely because the router data width is limited to a few choices. Likewise, when the PE changes from 128 bits and 400MHz to 128 bits * 200MHz, the router keeps the power higher when compared with Fig. 5.

Although it is possible to reduce the amount of buffers, the fact that the frequency remains fixed, prevents a finer adjustment by the router. Thus power is consumed unnecessarily.

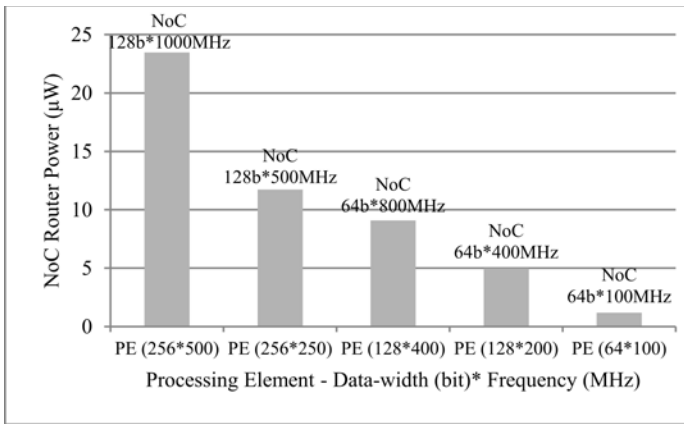


Figure 7: NoC Router Power – Frequency and Data Width Variation

C. Varying the frequency and data width

Varying both data width and frequency allows one to explore a larger design space. Thus, as shown in Fig. 7, the power can be reduced more significantly. This shows the great advantages found due to having greater freedom in defining the router's frequency and data width.

Taking the same examples discussed in subsections VI.A and VI.B, when the PE changes its frequency from 500MHz to 250MHz, the choices may be between 64 bits * 1000MHz, 128 bits * 500MHz or 256 bits * 250MHz. The choice with the lowest power, in this case, is to configure the router to 128bits * 500MHz. However, when the PE changes from 256 bits * 250MHz to 128 bits * 400MHz, the best configuration for the router is 64bit * 800MHz, different from that presented in Fig.5. This shows that the increased freedom provided by the reconfiguring both frequency and data width leads to further reductions in power consumption, when compared to any of the two approaches applied individually.

Table II shows the NoC power consumption attained for each of the considered processor configurations. It shows the results when considering only data width, only frequency and both variations at once. Note that only for the first two situations could a single parameter adjustment find the optimum result. For the remaining situations, both parameters had to be adjusted in order to find the best configuration able to minimize the power consumption without jeopardizing the performance.

TABLE II. BEST NOC POWER (MW) FOR EACH PROCESSOR CONFIGURATION

Proc. Element	NoC Freq. Variable	NoC Data Width Variable	NoC Width + Freq	Max Improvement over fixed
256b/500MHz	23.472	24.399	23.472	0.0%
256b/250MHz	11.736	11.736	11.736	50.0%
128b/400MHz	9.388	11.736	9.102	61.2%
128b/200MHz	4.979	6.474	4.694	80.0%
64b/100MHz	2.3472	1.285	1.195	94.9%

The rightmost column in Table II shows the attained reduction in power consumption when compared to a NoC that operates at the considered baseline configuration (128b * 1000MHz).

VII. CONCLUSIONS AND FUTURE WORKS

In this work we have presented the need for an adaptable Network-on-Chip when using reconfigurable processing elements. Whenever there is a processor able to dynamically modify its processing capabilities and/or a memory with adaptable bandwidth, the optimal NoC configuration able to supply the system need with minimal power will also change.

The presented case study considered an adaptable VLIW processor to identify the different throughputs that may be required from the NoC, and two different reconfiguration techniques were evaluated to find the lowest power consumptions. Results showed that the greatest reductions in power consumption are achieved only when both frequency and data width reconfiguration are applied in conjunction, with both techniques performing worse when applied individually.

These considerations and results open several possible future works. For example, algorithms to define individual frequencies and widths for each router, as well as the hardware mechanisms to support this feature, may lead to interesting gains, as the processors themselves are likely to be individually configurable. Other reconfiguration techniques for NoCs, such as buffers with variable depths or adaptable routing algorithms may lead to further gains when combined with the techniques presented here.

REFERENCES

- [1] F. Anjam, M. Nadeem, and S. Wong, "Targeting Code Diversity with Run-time Adjustable Issue-slots in a Chip Multiprocessor," in Design, Automation & Test in Europe, pp.1358–1363, 2011.
- [2] S. Wong, T. van As, and G. Brown, "p-VEX: A Reconfigurable and Extensible Softcore VLIW Processor", in *International Conference on Field-Programmable Technologies*, pp. 369–372, 2008
- [3] A.Malik, B.Moyer, and D. Cermak, "A Low Power Unified Cache Architecture Providing Power and Performance Flexibility", in *International Symposium on Low Power Electronics and Design*, pp. 241–243, 2000.
- [4] Anjam, F., Wong, S., Carro, L., Nazar, G. L., Rutzig, M. B., "Simultaneous Reconfiguration of Issue-width and Instruction Cache for a VLIW Processor", in *International Conference on Embedded Computer Systems: Architecture Modeling and Simulation*, July, 2012.
- [5] C. Zhang, F. Vahid, and W. Najjar, "A Highly Configurable Cache Architecture for Embedded Systems", in *International Symposium on Computer Architecture*, pp. 136–146, 2003.
- [6] D.H. Albonesi, "Selective Cache Ways: On Demand Cache Resource Allocation", in *International Symposium on Microarchitecture*, pp. 248–259, 1999.
- [7] T. Givargis and F. Vahid, "Tuning of Cache Ways and Voltage for Low-Energy Embedded System Platforms", in *Journal of Design Automation for Embedded Systems*, vol. 7, No. 1–2, pp. 35–51, 2002.
- [8] C. Zhang, F. Vahid, and W. Najjar, "Energy Benefits of a Configurable Line Size Cache for Embedded Systems", in *International Symposium on VLSI*, pp. 87–91, 2003.
- [9] Ahmad, B., A.T. Erdogan, S. Khawam. "Architecture of a Dynamically Reconfigurable NoC for Adaptive Reconfigurable MPSoC. in *Adaptive Hardware and Systems*", 2006. AHS 2006. First NASA/ESA Conference on 2006. Istanbul IEEE Press.405-411

- [10] Ahmad, B., T. Arslan. "Dynamically reconfigurable NoC for reconfigurable MPSoC in Custom Integrated Circuits", Conference, 2005. Proceedings of the IEEE 2005 2005. San Jose, CA IEEE Press. 277 - 280
- [11] Matos, D., Concatto C., Kreutz, M., Kastensmidt, F., Carro, L., Susin, A. "Reconfigurable Routers for Low Power and High Performance". In Very Large Scale Integration (VLSI) Systems, pp 2045-2057, 2011.
- [12] Rahmani, A. M., Liljeberg, O., Plosila, J., Tenhunen, H. "An efficient VFI-based NoC architecture using Johnson-encoded Reconfigurable FIFOs". In NORCHIP, pp. 1-5, 2010.
- [13] Rana, V., D. Atienza, M. D. Santambrogio, D. Sciuto, and D. G. Micheli, "A Reconfigurable Network-on-Chip Architecture for Optimal Multi-Processor SoC Communication" in 16th Annual IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC), 2008.
- [14] L. Benini and G. De Micheli, "Network on chips: A new SoC paradigm," IEEE Computer, vol. 35, no. 1, pp. 70–78, Jan. 2002.
- [15] C. Zeferino, M. Kreutz, and A. Susin, "RASoC: A router soft-core for networks-on-chip," in Proc. Conf. Des., Autom. Test Euro. (DATE), 2004, pp. 198–203.
- [16] Veale, B.F., Tull, M.P., and Antonio, J.K., "Dynamic Configuration Steering for a Reconfigurable Superscalar Processor," in 20th International Parallel and Distributed Processing Symposium, 2006. Apr. 2006.
- [17] Beck, A. C. S., Rutzig, M.B., Gaydadjiev, G., Carro, L., "Transparent Reconfigurable Acceleration for Heterogeneous Embedded Applications". DATE 2008, pp. 1208-1213.