A Fully Dynamic Reconfigurable NoC-based MPSoC: The Advantages of Total Reconfiguration

Paulo C. Santos[1], Gabriel L. Nazar[1], Fakhar Anjam[2], Stephan Wong[2],
Débora Matos[1], Luigi Carro[1]

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

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

Abstract. The growing demand for higher performance with limited energy dissipation evokes the need for an optimal use of system resources. In search of the best performance/power ratio, and considering that a static system is not able to optimally perform the various applications required by the user, runtime reconfigurability has become an avenue of research. One can find proposals for adapting dynamically processors, memories and their interconnections in order to provide a more suitable configuration for the system. However, there is no available study on the simultaneous reconfiguration for all these components in a system. This work presents a study where the reconfiguration of each different stage of a system - processing, memory and communication - is shown, proving the importance of the optimal use of each resource. Moreover, we show that a static component significantly undermines the potential power reduction achievable when reconfigurable components are used, being such adaptation at system level crucial to achieve elevate levels of energy reduction, while sustaining high performance. When all components are adapted in agreement, a total power reduction up to around 84% is possible.

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

1 Introduction

Multiprocessors System on Chip (MPSoCs) have emerged as a technology trend to provide the needs for growing performance and integration of different features in a single integrated circuit. Parallelism exploitation at different granularities has been crucial for the success found in these platforms. Processing Elements (PEs) that make up an MPSoC can consist of different architectures, like general purpose processors, VLIW processors, DSPs, vector processors and other accelerators. For effective

adfa, p. 1, 2011.

communication among processors and memories, Networks-on-Chip (NoCs) have been adopted, providing reusability, scalability, parallelism and high bandwidth [13].

However, the search in the design space is not only referent to performance, but also about consuming less power and energy. Thus, techniques able to significantly reduce the power consumption are essential for enabling the efficient use of aggressively scaled technologies.

Some techniques for reducing power and energy involve the reduction of frequency and voltage, as well know the DVFS (Dynamic Voltage and Frequency Scaling) strategy, 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 of instruction cache. 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 PEs of an MPSoC and the NoC routers is done, in several cases, using the memory addressing space of these PEs, as shown in Fig. 1. Thus, a reduction in the issue-width of a processor could reduce the bandwidth required from its memory, and consequently, the bandwidth of the NoC routers. Note that such changes also affect significantly the power consumption of the system. 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). Furthermore, as technology scales, the costs of interconnection wires becomes growingly significant. Thus, a fully reconfigurable system can find improved solutions in the design space.

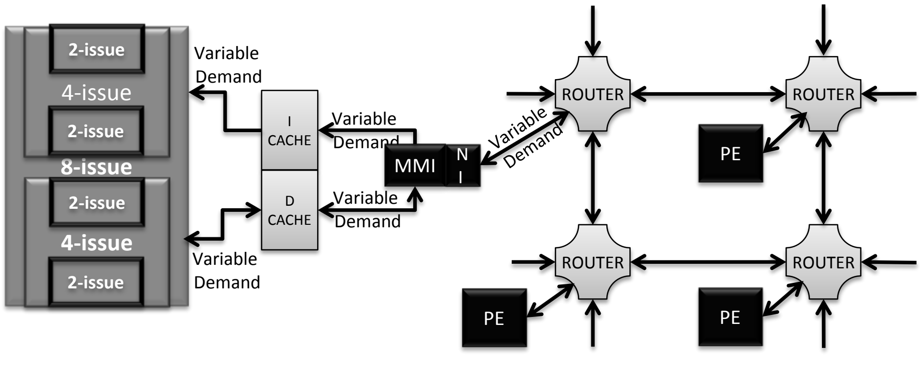

Fig.1. ρ-VEX processor, memories and interfaces

This article presents a study in which the main responsible of the system that contributes to performance and power consumption such processors, memories and system interconnection, can be reconfigured in search for better solutions. We show that partially adaptive systems are limited in the power reductions due to the power of the static parts. For example, a system that reconfigures memory and NoC but leaves the static processor will not be able to improve its power consumption beyond a certain limit. Most importantly, we quantify these limitations. Moreover, we exploit different means that may be used to dynamically modify the performance and power consumption of the system, namely scaling the operating frequency and partially powering off parts of the components, reducing bandwidth or issue width, for example.

This work is structured as follows. In section II we discuss the main related works. Section III shows how inefficient can be a partially reconfigurable system. 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 and section VII concludes the work, pointing out relevant future work possibilities.

2 Related Work

There are many works with the goal to reduce power consumption in processors that also try to minimize the performance penalty. Processors of different architectures and varied purposes were proposed using reconfigurability, enabling the development of dynamically reconfigurable processing elements [1][15][16].

In [15] the authors propose a processor where the reconfigurable logic works as an FU. The purpose of this processor is to increase the achievable 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. The reconfiguration of the superscalar processor is an attempt to increase ILP by executing as many instructions available, and it modifies the throughput required from the instruction memory. In [16] 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 proposed in [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 cache memory. In [4], experiments were made taking into account the need to change the instruction 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, the memory size and the size of the lines also brings changes in performance and power consumption, as shown in [8], where different results are obtained according to the line sizes.

However, for a system NoC-based with maximum configurability, including processor and memory, the communication must also be capable of reconfigure in order to follow the rest of the system. In the literature there are several studies regarding reconfigurable NoCs. In [9] 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 [10] the depths of the buffers are reconfigured at runtime, showing a significant reduction in power consumption without performance penalties. In [11] there is also a concern with the buffers of the NoC routers, where a novel reconfigurable FIFO using Johnson-encoding is used, reducing latency and the power consumed by the circuit. In [12] a way to dynamically reconfigure the switching and topology of the NoC at runtime is presented, reducing the critical path and latency.

None of the previous takes onto account the impact of having a reconfigurable component claiming other components to be reconfigurable as well in order to achieve minimum system energy. In this work, we show that significant gains can be achieved when one couples the reconfigurability of these three components (processor, memory and interconnection), quantifying the relevance of each one.

3 Inefficiency of the Partially Reconfigurable Systems

In a system with a processor and cache memories, if the processor presents techniques allowing reducing its power consumption, the power of the cache memories becomes proportionally even more important for the whole system. Therefore, techniques for reducing power consumption should also be applied to the cache memories, following the change in the processor, in search of the best performance/power ratio. For example, if a processor reduces its frequency from 400MHz to 100MHz, the cache memory can also reduce its operating frequency, since this 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 modification of these parameters directly influences 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 system bottleneck.

Exploiting this other issue, a NoC router which has a frequency of 400MHz and 128 bits of data width interconnected to a memory with 256 bits of data running at 200MHz works with maximum efficiency. But if this memory has the operation frequency 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. The worst cases occur when both frequency and data width are reconfigured in the PE.

4 Reconfigurable Parameters of the NoC Based System

In this section we present the reconfigurable parameters for a cache, a 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.

4.1 ρ-VEX VLIW Processor

The VLIW processor used was the ρ-VEX [2], extended with the contributions presented in [1]. This version of the ρ-VEX processor enables reconfiguring the VLIW processor in 2, 4 or 8 issues at runtime, and varying the frequency of operation. This directly affects the instruction memory bandwidth, since when varying the issue-width, the size of instruction also varies. 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 instruction throughput requirements, which modifies the optimal memory and communication configurations. In order to perform a complete analysis, the power consumption of the processor was extracted using the tool Cadence RTL Compiler Ultra.

Fig. 1 shows the base structure of the MPSoC, showing the ρ-VEX and the memory and router surrounding it. Note that it allows composing larger VLIW processors with independent 2-issue cores and that this modifies 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.

4.2 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 performance and power required. The size of the cache and cache lines, as well as associativity, varies in order to maintain the lowest power consumption without loss of performance, and hit ratio is assumed to be constant. In addition, the operating frequency of the memory was also varied, thus simulating a complete reconfigurable PE. We consider a cache that modifies its parameters and throughput dynamically along with the requirements of a reconfigurable processor. For the different configurations used in the instruction cache, we extracted the power consumption and area using the HP CACTI tool.

4.3 Network on Chip

The Network-on-chip used in this article is based on the SoCIN RaSoC router [14] with data width and frequency between 16 and 256 bits, and 100 and 1000MHz, respectively. For this analysis we extracted estimates of power through the Cadence RTL Compiler Ultra tool using a 65nm technology for NoC Router and HSpice for wires, using the Π model [17]. The RaSoC router allows different depths of buffers, but in this initial study this parameter was maintained fixed in 4 positions for each input channel. Another feature of the NoC used is that routers are full duplex, thus working with incoming links independent of outgoing links.

5 Experimental Setup

The experiment was based on the MPSoC shown in Fig. 1. A complete MPSoC tile shown in Fig. 2, formed by processor, memory and router was analyzed, where:

- The processor is capable of varying frequency and issue-width to maintain the lowest power consumption according to the application requirements.
- The instruction memory is able to vary the frequency, port size, cache size, line size and associativity, keeping the flow of data required by the processor with the lowest possible power consumption without loss of performance, allowing the variation of the maximum memory bandwidth required in each configuration.
- The NoC router is able to individually vary the frequency and the link data width (the number of wires connected to the router), enabling a variation of power consumed by the router and by the communication wires between routers.

Table I. Parameters used in the experiment

Parameters	Range	Initial Value
ρ-VEX issue-width	2/4/8	8
ρ-VEX frequency (MHz)	200/250/300/400/500	500
Cache port size (bits)	64/128/256	256
Cache frequency (MHz)	200/250/300/400/500	500
Cache Size (Kbytes)	8/16/32	32
Cache Line Size (Bytes)	16/32/64/128/256	256
Cache Associativity (Ways)	1/2/4/8	8
NoC Router frequency (MHz)	100 to 1000 – step 50	500
NoC Router data width (bits)	16/32/64/128/256	256
Default Setting Power Consumption (mW)	---	192.71

Based on extracted area provided by the tools, RTL Compiler and CACTI, and using the homogeneous arrangement of the components shown in Fig. 1, the length of the wires was estimated in 1.2 mm between the routers.

The experiment starts with the system using the initial values described in Table I and has its parameters changed seeking an optimal configuration of the entire system. Table I also presents the power consumed by the default setting, which will be compared with the reconfigurable system. The need to vary the frequency and/or issue-width of a processor is defined according to the application is running. This variation will dictate the need to vary the frequency and/or other parameters of the other components. For a more precise analysis, we enable reconfiguration initially only on the processor, then the processor and memory were reconfigured, and in the last step the systemwas fully reconfigured.

6 Analysis

In this section we study the impact of the reconfiguration of each element that composes a NoC-based MPSoC system (processor, memory and communication).

We observed that when we modify the bandwidth required by a processor, it is interesting that the memory and NoC are also reconfigured, especially in terms of frequency and data width, thus achieving a system with better performance/power. Fig. 2 illustrate the reference node of the MPSoC for the experiments. The processor initially runs at a 500MHz of frequency, with 8-issue (256bit instruction word). The memory runs with the configuration 8W32K256B-500M-256b (where 8W means 8-way associativity, 32K means 32KB cache size, 256B means 256 byte line size, 500M is the operating frequency in MHz and 256b means the data width in bits equivalent to processor issue-width). Finally, the NoC runs at 500MHz with 256bit data width. The MMI (Memory Management Interface) and the NI (Network Interface) are responsible for providing the interface between the PE and the NoC router.

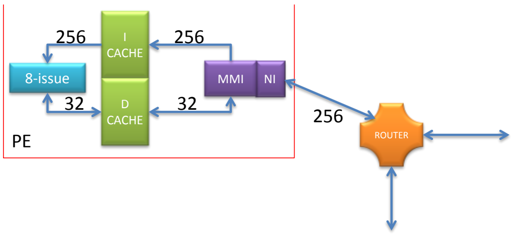

Fig. 2. MPSoC reference node

The purpose of this analysis is to quantify the impact of each reconfigurable component in MPSoC power consumption. We take as starting the initial value configuration shown in Table I and performed three tests, where the different configurations according to the parameters of column *Range* of Table I were applied.

6.1 Varying the processor parameters

Initially an application sets the reconfiguration of the processor, modifying the issue-width needed and the throughput from 256bit*500MHz (128Gbps) to 76.8Gbps, 64Gbps, 38.4Gbps, 32Gbps and 19.2Gbps, as shown in Fig. 3, respectively in columns 1 to 6. We can note in the first column, the results of power consumed in each element of MPSoC when the initial configuration is applied.

The application may also require a deactivation of some functional units in order to further reduce the power consumption, in this way the columns 3 to 6 of Fig. 3 shows the power consumed by the processor when these changes occur along with lowering of frequency and issue-width. The processor initially represented 45.6% of the power consumed by the MPSoC, being this value only 10.2% when reconfigured for ρ-VEX 2 issue 300MHz. We can also observe that the power consumption in the other components remains static, limiting the attainable gains. Even when the processor power consumption is reduced 7.37 times (from the first to the last column), the entire system power is only reduced 39.45%.

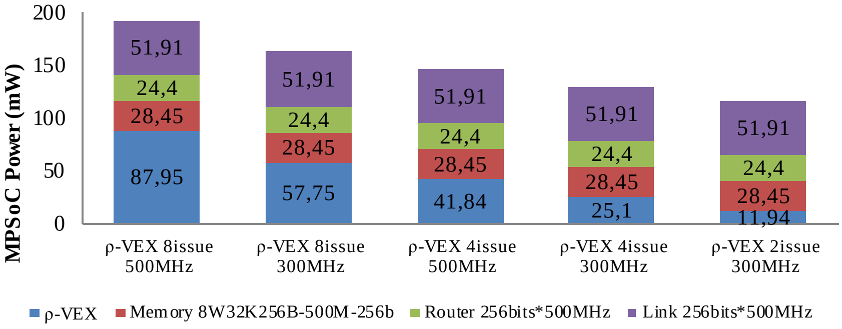

Fig. 3. Power results (mW) – Reconfigurable Processor

6.2 Varying the processor and memory parameters

Fig. 4 shows the results of power consumption for an MPSoC tile where the processor and memory are reconfigured concordantly (processor and memory operate at same frequency). The first column shows the power consumption on the default system. The columns in Fig. 4 show the changes occurring in memory frequency, according to the processor reconfiguration.

Note that, as the power from the processor and the memory is reduced, the NoC power, due to the router and associated wires, becomes a limiting factor. When the power of the processor is reduced 7.37 times and the memory power is reduced 4.06 times, the overall reduction is limited to 50.58%, due to the NoC static nature.

6.3 Concurrently varying the Processor, Memory and NoC parameters

When we vary the frequency and data width of the router, its power consumption varies, as well as that of the router wires. When we vary the data width of the NoC, there is also the possibility of powering off the buffers associated with the unused channel bits, leading to significant reductions in the total NoC power.

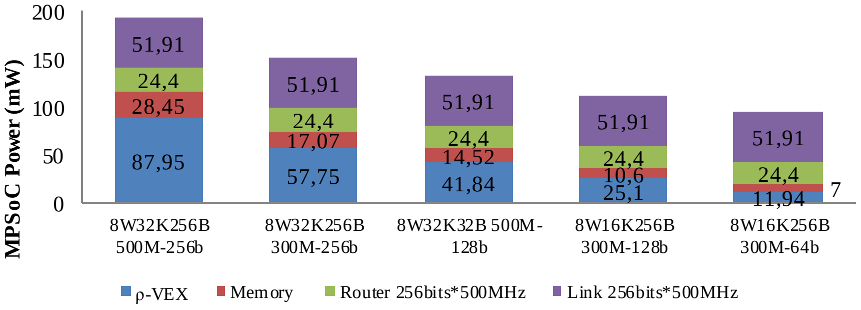

Fig.4. Power results (mW) – Reconfigurable Processor and Memory

Fig. 5 shows the total power consumed by a fully adaptive system. Note that, when all elements are adapted to maintain compatible capabilities, the maximum power reductions are achievable. The total power reduction attainable by such a system is 84.12%, as there is no static component to limit the gains.

7 Conclusions and Future Works

In this work we have presented and quantified the need for adaptability in all components that contribute significantly to system power consumption. 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 needs with minimal power will also change. Thus, leaving any of the system components with exceeding power or insufficient performance will make such a component either a source of energy waste or performance bottleneck.

The presented case study considered an adaptable VLIW processor, coupled to an equally adaptable memory and NoC infrastructure. Results showed that the greatest reductions in power consumption are achieved only when all system elements are adapted concordantly. We have shown that a system with an adaptable processor can only reduce power by 39.45%, while a fully adaptable MPSoC can, for example, reduce its power by 84.12% according the parameters considered in this paper. Quantization of these gains is crucial to guide future research directions.

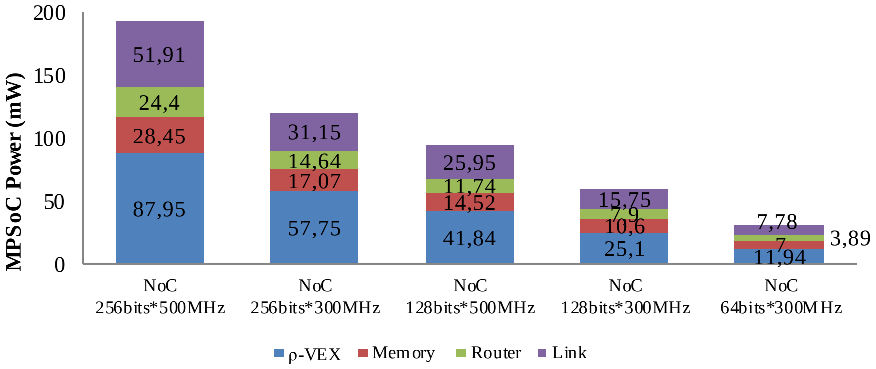

Fig.5. Power results (mW) – Fully adaptable system

These considerations and results open several possible future works. For example, to consider heuristics to coordinate the optimum system configuration, taking into account all relevant layers at once, may successfully exploit the parameters presented. Furthermore, other reconfiguration directions, such as NoC routing algorithms or the processors register file size may be added to the design space exploration.

8 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, “ρ-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 Enbedded Computer Systems: Achitecture 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 Istanbul 2006. IEEE Press.405-411
10. 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.
11. 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.
12. 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.
13. L. Benini and G. De Micheli, “Network on chips: A new SoC paradigm,” IEEE Computer, vol. 35, no. 1, pp. 70–78, Jan. 2002.
14. 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.
15. 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.
16. Beck, A. C. S., Rutzig, M.B., Gaydadjiev, G., Carro, L., “Transparent Reconfigurable Acceleration for Heterogeneous Embedded Applications”. DATE 2008, pp. 1208-1213.
17. Sakurai T. Approximation of wiring delay in mosfet lsi. In IEEE Journal of Solid-State Circuits, volume 18, pages 418–426, 1983.