

PIRATE: A Framework for Power/Performance Exploration of Network-On-Chip Architectures

Gianluca Palermo Cristina Silvano

Dipartimento di Elettronica e Informazione
Politecnico di Milano, Milano, Italy
{gpalermo, silvano}@elet.polimi.it

Abstract. In this paper, we address the problem of high-level exploration of Network-on-Chip (NoC) architectures to early evaluate power/performance trade-offs. The main goal of this work is to propose a methodology supported by a design framework (namely, *PIRATE*) to generate and to simulate a configurable NoC-IP core for the power/performance exploration of the on-chip interconnection network. The NoC-IP core is composed of a set of parameterized modules, such as interconnection elements and switches, to form different on-chip micro-network topologies. The proposed framework has been applied to explore several network topologies by varying the workload and to analyze a case study designed for cryptographic hardware acceleration in high performance web server systems.

1 Introduction

Designing complex System-On-Chip (SoC) solutions, such as multi-processors (MPs) or network processors, requires a flexible platform-based approach for both hardware and software sides of embedded architectures. The growing diffusion of MP-SoC embedded applications based on the platform-based design approach requires a flexible tuning framework to assist the phase of *Design Space Exploration (DSE)*. The overall goal of the DSE phase is to optimally configure the parameterized MP-SoC platform in terms of both energy and performance requirements depending on the target application. MP-SoC hardware platforms are usually built around a *network centric* architecture interconnecting a set of processors and IP (Intellectual Property) cores to the memory subsystem and I/O interfaces. The communication infrastructure can connect up to tens of master and slave IP nodes through the network architecture. In this scenario, the target architecture is based on the *Network-on-Chip (NoC)* approach [1], where on-chip communication represents the core of the overall system design methodology and it can be specified somewhat independently of the modules composing the platform. The design of high-performance, reliable, and energy efficient interconnect-oriented MP-SoCs raises new challenges in terms of design methodologies. A simulator to estimate performance, while varying network parameters, faces only one aspect of the architectural exploration at the system-level. To efficiently support system-level architectural exploration, we

need a high-level simulator and a power estimator to dynamically evaluate accurate power/performance trade-offs. This is especially necessary because the interconnection network accounts for a significant fraction of the whole energy consumed by the SoC, and this fraction is expected to grow in the next future [2] due to the growing complexity of packet routing and transaction management policies.

In this paper, we address the problem of the power/performance exploration of the NoC architectural design space at the system-level. The main goal is to support the exploration and optimization of network-centric on-chip architectures from the early stages of the design flow, when only the high-level description of the system can be used to simulate the timing and power behavior, and any other detailed information of the system will be known after the synthesis and optimization phases.

In particular, we propose a methodology to generate and to simulate a configurable NoC-IP architecture to support the efficient yet accurate exploration of the interconnection system of a SoC. Starting from the system-level specification of the interconnection network, the proposed framework supports the generation of a configurable system-level model described in SystemC to dynamically profile the given application. A set of power models are dynamically plugged-in in the simulator to provide accurate and efficient power figures for the different architectures.

The proposed framework has been used to explore two different design scenarios. First, we discuss the results obtained by the exploration of different NoC topologies by varying the workload. Second, we present the results obtained by applying the proposed methodology to a case study: An on-chip cryptographic accelerator for high performance web server systems.

The rest of the paper is organized as follows. A review of the most significant works appeared in literature to support the design of NoC architectures is reported in Section 2. The proposed design exploration framework is described in Section 3. Section 4 discusses the experimental results carried out to evaluate the accuracy and efficiency of the proposed framework, while some concluding remarks and future directions of the work have been outlined in Section 5.

2 Background

Many on-chip micro-network architectures and protocols have been recently proposed in literature, in some cases focusing on specific system configurations and application classes. The *SPIN Micronetwork* [3] represents a on-chip micro-network based on deterministic routing. The *Silicon Backplane Micronetworks* [4] based on a shared-medium bus and time-division multiplexing offers an example of transport layer issues in micro-network design. The *Octagon* on-chip communication architecture for OC-768 Network Processors has been proposed in [5].

The design of high-performance, reliable, and energy efficient interconnect-oriented MP-SoCs raises new challenges in terms of design methodologies ([1], [6]). Low-level power estimation tools (such as *Synopsys Power Compiler*) require synthesized RTL descriptions. Although these tools provide good levels of accuracy, they require long simulation time, becoming an unfeasible alternative for complex MP-SoCs. The exploration of MP-SoCs requires architectural-level power and performance simulators such as *SimpleScalar* [7], *Wattch* [8], and *Platone* [9]. A set of system level power optimization tools have been presented in [10].

In this direction, the *Orion* approach [11] consists of a power-performance interconnection network simulator to explore power-performance trade-offs at the architectural level. The framework can generate a simulator starting from a micro-architectural specification of the interconnection network. In the *Orion* approach, a set of architectural-level parameterized power equations have been defined to model the main modules of the interconnection network (such as FIFO buffers, crossbars, and arbiters) in terms of estimated switch capacitances to support dynamic simulation.

Power models for on-chip interconnection architectures have been recently proposed in literature. A survey of energy efficient on-chip communication techniques has been recently presented in [2]. Techniques operating at different levels of the communication design hierarchy have been surveyed, including circuit-level, architecture-level, system-level, and network-level. Patel *et al.* [12] focused on the need to model power and performance in interconnection networks for multiprocessor design, and they proposed a power model of routers and links. An estimation framework to model the power consumption of switch fabrics in network routers has been proposed in [13]. The work introduces different modelling methodologies for node switches, internal buffers and interconnect wires for different switch fabric architectures under different values of traffic throughput. The proposed modelling and simulation framework is limited only to switch fabrics. More recently, the same authors introduced in [14] a packetized on-chip communication power model for multiprocessors network design.

3 PIRATE NoC-IP Architecture

PIRATE is a Network-on-Chip IP core composed of a set of parameterizable interconnection elements and switches connected by different topologies. The switch architecture is based on a *crossbar topology*, where the $(N \times N)$ network connects the N input FIFO queues with N output FIFO queues. The number of each input (output) FIFO queue is configurable as well as queue length. The incoming packets are stored in the corresponding input queue, then, the packets are forwarded to the destination output queue. The I/O queues and the crossbar are controlled by the Switch Controller, that includes a static routing table and the arbitration logic. The *PIRATE* switch is a synchronous element requiring a one-cycle delay for each hop. The switch is based on wormhole routing and static routing defined by a routing table customized by the designer.

The design of *PIRATE* NoC-IP is supported by an automatic framework.

3.1 PIRATE Design Framework

The main goal of the *PIRATE* framework is the exploration of the design space to define the optimal configuration of the interconnection system for the target application. The *PIRATE* framework is mainly composed of the following modules:

- Generator of synthesizable Verilog RTL model for the configurable NoC-IP core;
- Automatic power characterization flow;
- Cycle-based SystemC simulation model for dynamic profiling of power and performance.

Synthesizable RTL Generator This module consists of an automatic generator of the Verilog RTL description of the generic *PIRATE* NoC configuration. The module receives as input a configuration file describing the micro-architectural parameters and the structure of the network. In the configuration file the designer can specify the following set of parameters:

- *Switch Architecture*: The switch architecture is parametric in terms of number and length of I/O queues and interconnection width;
- *Network Topology*: A set of parameters defines the interconnection topology of the switches in the network. The explored standard network topologies are: Ring, Double Ring, Mesh, Cube, Binary-Tree, and Octagon. Other network topologies can be generated *ad hoc* to optimize the target architecture;
- *Connections Encoding*: The information flowing through the network can be encoded. Using this set of parameters the designer can insert standard encoding/decoding techniques;
- *Information Flow in the NoC*: The routing mechanism in the switches is based on a static routing table. The designer can customize the routing table for each switch to balance and to optimize the traffic in the network.

Power Characterization Flow A methodology to automatically create the power models of the *PIRATE* NoC-IP has been defined and implemented to dynamically profile at system-level the power behavior of each module of the interconnection system. The methodology is based on the power characterization of each parameterized module of the interconnection system (i.e. switch, encoder/decoder, etc.). For each module, at the end of the characterization flow, we automatically generate the corresponding analytical model based on its design space parameters and the traffic information derived from simulation.

We define the *design space* as the set of all the feasible architectural implementations of a module. A possible configuration of the target module is mapped to a generic point in the design space as the vector $a \in \mathcal{A}$, where \mathcal{A} is the architectural space defined as $\mathcal{A} = S_{p_1} \times \dots \times S_{p_l} \dots \times S_{p_n}$ where S_{p_l} is the ordered set of possible configurations for parameter p_l and " \times " is the cartesian product. To exemplify our method, let us consider the power model of the switch. According to [13], our power model of the switch is dependent on traffic and design space parameters:

$$P_{SW}(\mathcal{A}, TR) = B_0(\mathcal{A}) + B_1(\mathcal{A}) \times TR \quad (1)$$

where TR is the traffic factor, \mathcal{A} is the architectural space defined above and B_0, B_1 are the model coefficients.

Due to its configurability, the NoC IP core spans a very large design space \mathcal{A} and this causes the impossibility to characterize each configuration, since each point to be characterized requires an RTL synthesis and a variable number of gate-level simulations and power estimations. To reduce the number of configurations to be characterized, our approach is based on the fundamental theories of the statistical design of experiments (*DoE*) [15] [16].

Figure 1 shows the proposed automatic power characterization flow. It is based on the standard *Synopsys* [17] gate-level power estimation flow (dark grey) and it is composed of three main phases:

- *Design of Experiments*: This phase concerns the planning of the experiments before synthesis and simulation of each module. It consists of the sampling of the design space and the automatic generation of the RTL description of the target module and the corresponding testbench for the following simulation phase.
- *Standard Power Estimation Flow*: This phase represents the core of the characterization methodology and it is based on the *Synopsys* tool chain. The standard flow receives as input the RTL description of the elements generated and, by using the *Synopsys Design Compiler* tool and the target technology library, the flow performs the synthesis of the corresponding gate-level description. The value for the power dissipation of the element under analysis can be obtained after the gate-level *VCS* simulation (by using the input patterns automatically generated in the previous phase of the flow) and the power estimation by using Design Power.
- *Empirical Model Building*: This phase generates the power model of the interconnection elements. In the power model characterization phase, the values of the coefficients of the high-level model are computed by linear regression over the set of experiments given during the *DoE*. The phase receives as input the power estimation values of a point in the design space for the given stimuli. By using this information, the flow can choose to re-simulate and to re-estimate

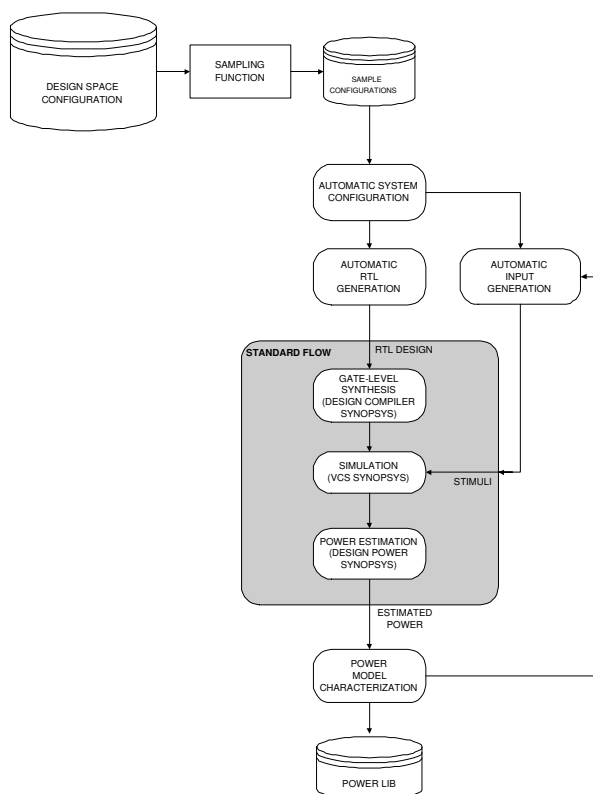


Fig. 1. The proposed power characterization flow

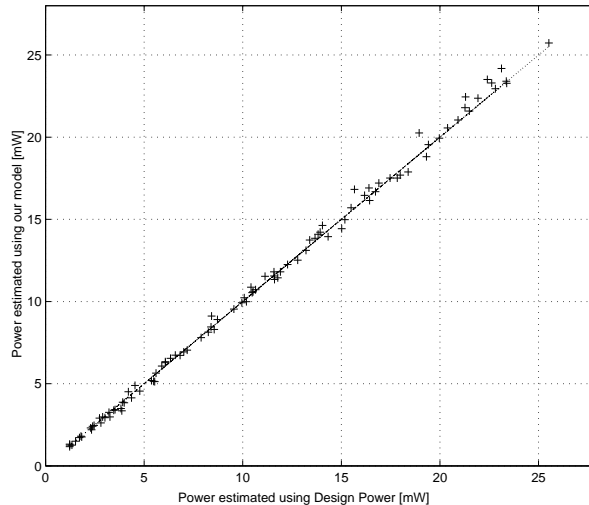


Fig. 2. Scatter plot of power estimated by our model with respect to power estimated by Synopsys Design Power for the switch design space.

the power of the element with other input stimuli to obtain a more accurate modelling [16]. The derived power model is then inserted in the power library.

The proposed methodology is technology-dependent, thus providing very accurate models. Currently, the STMicroelectronics 0.18 μm HCMOS8 technology is used. The validation results are shown in Figure 2 to evaluate the accuracy of the power model of the switch given in Equation (1). The scatter plot compares the power estimates obtained by our method and those obtained by using *Synopsys Design Power*. In the scatter plot, the points are very close to the diagonal, with a standard deviation within 5%.

Cycle-based Simulation Model The *PIRATE* NoC-IP simulation model has been developed at system-level by using SystemC 2.0 [18]. The model has been developed at Bus Cycle Accurate (BCA) level, therefore providing cycle-based accurate timing information and power estimates. The model is completely configurable in terms of NoC micro-architectural parameters, such as network topology, number of masters/slaves for each switch, length of input/output queues for each switch. Basically, the simulator receives as input the same configuration file used to create the synthesizable RTL description of the network and it generates the simulatable interconnection module described in SystemC.

The power models obtained by the automatic characterization flow presented above have been plugged in the system-level simulation model to perform a fast power-performance exploration of the NoC system based on dynamic profiling. From one side, the system-level simulation can guarantee the *efficiency* during the exploration phase, from the other side, the plug-in of power models estimated at gate-level can guarantee the *accuracy*. The network simulator can support two types of power analysis: *Static Analysis*, providing the total average power for the target application, and *Dynamic Analysis*, providing a cycle-accurate power and performance profiling.

4 Experimental Results

In this section, we show how the *PIRATE* framework can be used for a fast system-level exploration of power performance trade-offs of on-chip micro networks. Two different exploration scenarios have been analyzed.

First, we discuss the experimental results obtained by the exploration of the power/performance effects of different traffic patterns to the parameterizable NoC-IP. The exploration compares different network topologies. Second, we present the results obtained by applying the proposed methodology to the exploration of the CryptoSoC case study: An on-chip cryptographic accelerator for high performance web server systems based on SSL/TLS protocol. CryptoSoC has been developed in the MEDEAplus A304 project. In this case, the exploration analyzes power/performance trade-offs for different architectural configurations of the modules composing the NoC-IP, given the representative workload of the target network application.

4.1 Exploration of NoC Topologies

The average power and latency associated with the parameterizable NoC-IP have been analyzed by varying the packet injection rates. We simulated and compared five different network topologies connecting 8 nodes throughout the NoC-IP: Octagon, Cube, Double-Ring, Mesh, and Binary-Tree. For all topologies, the simulator generates uniformly distributed random traffic: each node injects packets in the network uniformly to random node destinations following the Poisson distribution for the injection time.

For different network topologies, Figure 3 and Figure 4 respectively show the average packet latency and the average network power with respect to the average packet injection rate. In both figures, a higher number of sample points for high values of average packet injection rates have been evaluated to find the saturation points with more accuracy. A similar trend can be noted for all topologies, but with a different behavior *before* with respect to *after* the corresponding saturation point. The behavior *before* saturation shows the average packet latency slightly increasing as the workload varies from 0 to 0.2, and almost constant for workload from 0.2 to 0.75. The average network power increases proportionally with the workload in the range from 0 to 0.75. *After* the saturation point, the average packet latency grows rapidly, while the average power remains constant or slightly decreases, because the network cannot handle a higher packet injection rate. Octagon and Cube topologies outperform the other topologies in terms of both latency and power for all workload values, and they tend to saturate for workload values close to 1. The Binary-Tree presents the early saturation point for latency (around 0.75), and Double-Ring and Mesh topologies also early saturate (around 0.85 and 0.9 respectively). A similar behavior can be noted for the average power.

4.2 Exploration of CryptoSoC Architecture

The CryptoSoC architecture (shown in Figure 5) is composed of the following modules: Public Keys (PK) Cache, Sessions Keys (SK) Cache, Initialization Vectors (IV) Cache, Hashing Function State (ST) Cache, Hashing Module SHA, Cryptographic Module RC4, RSA Module, and the Packet In/Out Buffers to interface

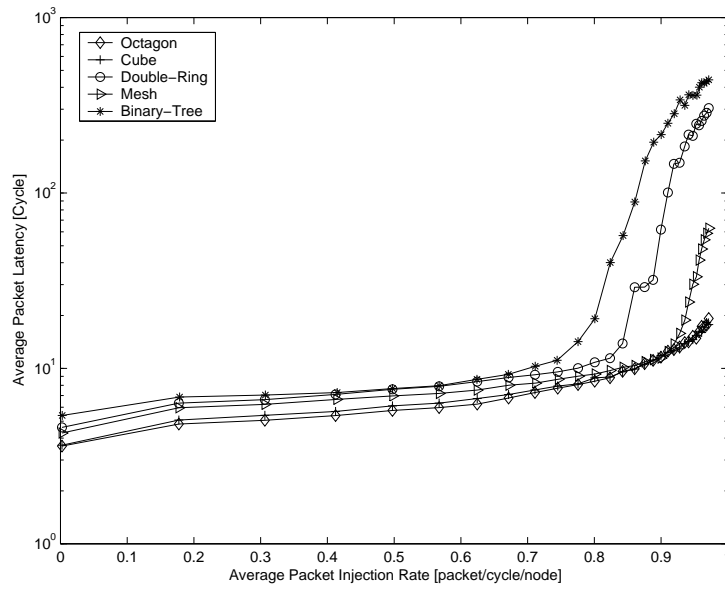


Fig. 3. Average packet latency with respect to average packet injection rate for different network topologies of the *PIRATE* NoC-IP

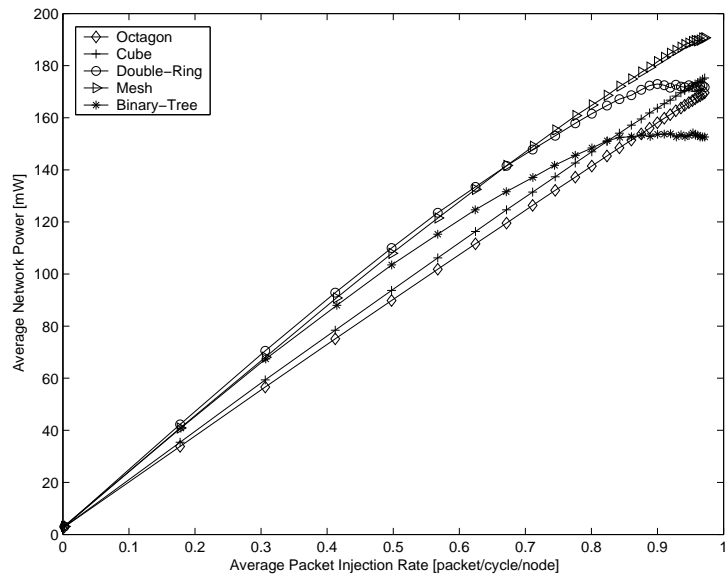


Fig. 4. Average network power with respect to average packet injection rate for different network topologies of the *PIRATE* NoC-IP

the PCI System Bus. In this case, the exploration analyzes power/performance trade-offs for different micro-architectural network topologies given a representative workload of the target network application. Furthermore, we explored a custom network topology designed for the CryptoSoC architecture based on a dynamic profiling of the application information flow to eliminate switch congestions.

Table 1 reports the average packet latency and the average network power for different network topologies, given the workload for the target application. The analyzed standard topologies are Octagon, Cube, Double-Ring, and Mesh, while the last row reports the *ad hoc* optimized topology. The *ad hoc* topology slightly outperforms the other standard topologies for both average network power and packet latency. Among the standard topologies, the Double-Ring topology presents the lower average network power due to its simple structure. Due to the low workload of the target application, the average packet latency of the Double-Ring is similar to the more complex Octagon topology.

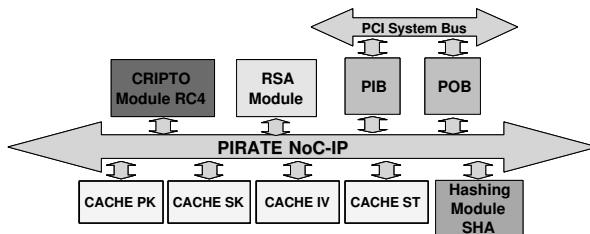


Fig. 5. The CryptoSoC Architecture

Table 1. Comparison of different network topologies for the CryptoSoC architectures in terms of average packet latency and average network power.

Topology	Avg. Packet Latency [Cycle]	Avg. Network Power[mW]
Octagon	2.80	16.19
Cube	3.22	16.30
Double-Ring	2.87	14.92
Mesh	2.92	16.18
Ad-Hoc	2.10	14.05

5 Conclusions

In this paper, we propose a design framework to generate and to simulate a configurable NoC-IP core to support the fast exploration of the on-chip interconnection network for MP-SoC applications. The NoC-IP is composed of a set of a parameterized switches to form different network topologies. The proposed framework has been applied to explore the power/performance effects of different different network topologies at varying traffic patterns. Furthermore, we applied the proposed framework to a case study, an on-chip cryptographic hardware accelerator for high

performance web server systems. Future work is directed towards the plug-in of the *PIRATE* NoC-IP into a high-level domain-specific flexible MP-SoC platform oriented towards networking applications and supporting quality of service.

References

1. L. Benini and G. De Micheli. Networks on chips: A new soc paradigm. *Computer*, pages 70–78, January 2002.
2. M.B. Srivastava V. Raghunathan and R.K. Gupta. A survey of techniques for energy efficient on-chip communication. In *Proc. of DAC-40: ACM/IEEE Design Automation Conference*, pages 900–905, June, 2003.
3. A. Greiner L. Mortiez C. A. Zeferino A. Adriahtenaina, H. Charlery. Spin: A scalable, packet switched, on-chip micro-network. In *Proc. of DATE-2003: Design, Automation and Test in Europe Conference and Exhibition*, March, 2003.
4. www.sonicsinc.com.
5. S. Dey F. Karim, A. Nguyen and R. Rao. On-chip communication architecture for oc-768 network processors. In *Proc. of DAC-38: ACM/IEEE Design Automation Conference*, June, 2001.
6. A. Mihal K. Keutzer S. Malik J. Rabaey A. Sangiovanni-Vincentelli M. Sgroi, M. Sheets. Addressing the system-on-a-chip interconnect woes through communication-based design. In *Proceedings of the 38th Design Automation Conference*, June 2001.
7. D. Burger, T. M. Austin, and S. Bennett. Evaluating future microprocessors: The simplescalar tool set. Technical Report CS-TR-1996-1308, University of Wisconsin, 1996.
8. D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. In *Proc. of ISCA-00*, pages 83–94, 2000.
9. Tony D. Givargis and Frank Vahid. Platune: A tuning framework for system-on-a-chip platforms. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 21(11):1317–1327, November 2002.
10. L. Benini and G. De Micheli. System-level power optimization: techniques and tools. *ACM Transactions on Design Automation of Electronic Systems.*, 5(2):115–192, January 2000.
11. L.S. Peh H.S. Wang, X. Zhu and S. Malik. Orion: A power-performance simulator for interconnection networks. In *MICRO-35: Int. Symposium on Microarchitecture, Instambul, Turkey*, Nov., 2002.
12. S. Yalamanchili C. Patel, S. Chai and D. Schimmel. Power constrained design of multiprocessor interconnection networks. In *Proc. of ICCD*, pages 408–416, Oct., 1997.
13. T. T. Ye, L. Benini, and G. De Micheli. Analysis of power consumption on switch fabrics in network routers. In *Proc. of DAC-39: ACM/IEEE Design Automation Conference*, June, 2002.
14. L. Benini T. T. Ye and G. De Micheli. Packetized on-chip interconnect communication analysis for mpsoc. In *Proc. of DATE-2003: Design, Automation and Test in Europe Conference and Exhibition*, March, 2003.
15. Douglas C. Montgomery. *Design and Analysis of Experiments, 5th Edition*. John Wiley & Sons, New York, 2001.
16. G. E. Box and N. R Draper. *Empirical Model-Building and Response Surfaces*. John Wiley & Sons, New York, 1987.
17. www.synopsys.com.
18. *SystemC 2.0 User's Guide*. 2002.