# A TLM-BASED NETWORK-ON-CHIP PERFORMANCE EVALUATION FRAMEWORK

*Johanna Sepúlveda, Gino Lozada, Marius Strum*

Integrated Systems Group - GSEIS
Electronics Systems Department
Microelectronics Laboratory
São Paulo University – USP

{jsepulveda, strum}@lme.usp.br

**ABSTRACT**

The variety of interconnection structures presently nowadays for *SoC* (System-on-Chip), bus and networks-on-Chip *NoCs*, each of them with a wide set of setup parameters, provides a huge amount of design alternatives. Although the interconnection structure is a key *SoC* component, there are few design tools in order to set the appropriate configuration parameters for a given application at the firsts stages of the *SoC* design. The present study aims to enhance the *SoC* communication platform design area, when a Network-on-Chip *NoC* is used. The novelty of this work is the performance evaluation of different *NoC* configuration at firsts stages of the *SoC* design flow (SystemC TLM). For evaluating the *NoC* a monitoring process was carried on. Our approach employed a set of metrics for *NoC* to elucidate the global and inner *NoC* behavior under a wide variety of traffic conditions.

## 1. INTRODUCTION

Electronic System design is being revolutionized by the widespread adoption of the system-on-chip (*SoC*) paradigm. The *SoC* is a computational system integrated into a single chip. The *SoC* designers are faced with the task of meeting the design requirements in a reduced time-to-market. Such scenario promoted an IP-based *SoC* design methodology that allows the reuse of key *SoC* functional components.

The *SoC* design is characterized by two design strategies: “meet-in-the-middle” that combines top-down and bottom-up design flows and “orthogonality” between the *SoC* communication and computation platforms. System design methodologies benefit from these two dimensions. The communication platform can be implemented using one out of two possible structures: bus or network-on-chip (*NoC*). On a bus structure, all the computation devices share the same transmission medium. The advantages of the shared-bus platforms (single or hierarchical) are simple topology and low area cost[1]. However, one of the problems associated with the growing size of *SoC* design arises from nonscalable global wire delays. As the bus length and the number of IP blocks increases, the delay associated to data transmission may become too large.

*NoCs* are becoming attractive communication structures for *SoC*s whose computation platform integrates a great number of IPs. A *NoC* is an integrated network that uses routers to allow the communication between the computation platform components. According to [1], the *NoC* appears as an alternative structure to overcome the hierarchical buses restrictions due to the following characteristics: (i) Bandwidth scalability; (ii) energy efficiency; (iii) distributed routing decision. However, *NoCs* are complex and costly when compared to buses [2]. The communication platform has a significant impact on the *SoC* performance. The platform type selection (bus or *NoC*) is carried out according to the system requirements. This paper addresses the design of a *NoC*-based communication platform.

A *NoC* may be configured by a set of physical (topology, channel structure and routers structure) and logical (routing and arbitration mechanisms) configuration parameters. The *NoC* platform design requires a constructive/tuning process of the configuration parameters in order to create a *NoC*-based platform *instance* that meets the *SoC* communication requirements and optimizes a set of performance metrics. In order to be efficient, an exploration phase among the possible instances, each defined by a set of configuration parameters must occur as early as possible along the *SoC* design flow. Different instances can be evaluated comparing their resulting performances.

The performance evaluation process can be made during three different *SoC* design flow phases. At the end of the design flow, from a *SoC* prototype, guaranteeing the evaluation accuracy. In the middle of the design flow, from a RTL model, achieving a compromise between accuracy and development time. At the beginning of the design flow from a Transaction Level Model-TLM model that is faster but less accurate than the RTL model.

In this paper we show that, despite its lower accuracy, high fidelity decisions concerning the search

of an optimal *NoC* instance can be taken from performance estimations made from a high abstraction TLM model. Previous works carried out performance estimation employing either RTL models [3-6] or TLM models [7]. In all cases estimations were made using throughput and latency as performance indicators. Such global performance parameters limit the designer's decision possibilities because they do not provide information about the inner characteristics of the *NoC*. This paper proposes a TLM-based *NoC* simulation and evaluation framework composed of traffic generators that supports a large variety of traffic conditions and a monitoring tool that annotates the events of the communication model required to evaluate the global as well as the inner network performance. Our experimental results show that the framework is able to quickly produce high fidelity results exploring a very large design space.

The text is divided in five sections. Section 2 presents an overview of the previous *NoC* performance evaluation works. Section 3 explains the *NoC* design principles. Section 4 describes the proposed framework. In section 5 the simulation results are presented. Finally we present our conclusions in section 6.

## 2. RELATED WORK

Jalabert et al. [3] and Lemaire et al. [4] presented the performance evaluation of a *NoC* using a RTL (Register Transfer Level) model. They employed two performance metrics: throughput and latency in order to study different topology alternatives. Pande et al. [5] proposes a RTL evaluation methodology to compare the performance of a variety of *NoC* topologies. The work uses the throughput and latency to evaluate the performance of six *NoC* instances. In order to estimate the *NoC* area, they synthesized the VHDL models using a standard cell-based, CMOS 0.13 μm technology. Bolotin et al. [6] present *QNoC*, a RTL-based *NoC* with *QoS* features. This work implemented besides the best effort service, a guaranteed throughput service. In order to evaluate the performance, two communication patterns are adopted: a uniform pattern, where all nodes communicate in a uniform way and a non-uniform pattern, where the communication between closest routers is higher. Few works addressed the performance estimation problem from higher abstraction level models. Dumitrascu et al. [7] conducted the performance evaluation of an application specific communication platform from a TLM model using throughput and latency as performance metrics. In order to expand the designer's possibilities to select a *NoC* instance that best suits the system's communication requirements, the set of performance metrics should provide the global as well as the inner behavior/performance of the *NoC*.

## 3. *NOC* DESIGN

The communication platform design flow shown in Figure 1 is composed of 3 phases: 1) Platform type selection (*NoC* or bus); 2) *NoC* configuration and 3) *NoC* verification process. The focus of this paper is on the phases 2 and 3. In order to perform the *NoC* configuration phase, the designer has to use information defined in previous design tasks: 1) Computation platform composition (*IP* type and quantity); 2) Library of communication components; and (3) Component mapping, where the number of routers and their connection to the IP computation components are defined. The *NoC* configuration phase is carried out in two stages: *physical stage*, where the *NoC* physical characteristics are configured (topology, channel structure and routers structure) and *logical stage*, where the information flow is established (routing and arbitration mechanisms). The *NoC* verification phase checks the performance adherence to the communication requirements. If the *NoC* instance does not meet the communication requirements a parameter tuning is carried on.

### 3.1 *NoC* communication model

The information through a *NoC* flows as packets. Packets are composed of headers, trails and payload of arbitrary size. A packet can be decomposed into smaller sized information called *flits*. The exchange of information between any pair of *IPs* connected to the *NoC* may be modeled as a transaction. A transaction is defined as the set of commutations (switch information at each router) required to complete the communication between a master-slave pair. In our work, a transaction is modelled as the set of events shown in figure 2. This model allowed us to analyse both the outer as well as the inner behavior/performance of the *NoC*. The transaction of figure 2 is composed of three commutations at routers *r*, *r+1* and *r+2*. Each line represents a packet commutation composed of the events: store (STO), arbitration process (ARB), routing process (ROT), sent header *flit* (HDR), sent packet size *flit* (SZ) and sent the set of payload *flits* (PAY1, PAY2).

## 4. FRAMEWORK

The proposed framework is a simulation-based approach that employs a TLM-SystemC abstract model to evaluate the *NoC* performance shown in figure 3. The system is composed of a set of master blocks that communicate with a set of slave blocks through a *NoC*. The communication process is monitored and evaluated by a tool that collects and analyzes the communication events. A set of performance metrics is calculated and used to carry out the *NoC* performance estimation for a wide variety of traffic conditions. The *NoC* framework allows to compare a wide number of *NoC* instances, each defined by a set of configuration parameters.

### 4.1. Traffic Generator and Traffic Receptors

The traffic generators are used to emulate the behavior of the master *IPs*. They establish different traffic conditions. In order to achieve a broad "evaluation coverage" four different types of traffic

should be combined: (i) application specific; (ii) corner cases; (iii) parametric; (iv) pseudo-random. In this work we only used types (iii) and (iv). The traffic receptors emulate the behaviour of the slave *IPs*. They confirm the reception of the correct information.

### 4.2. Network-on-Chip

The framework employs a cycle accurate timed TLM model of a parametrical *NoC* (Hermes_Temp)[1] that implements a packet-switched communication and wormhole flow control. A *NoC* instance is specified by

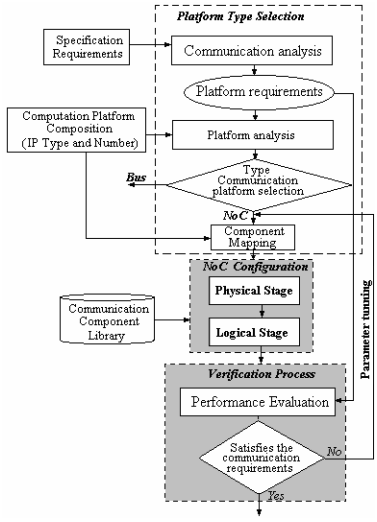


Fig. 1 Communication platform design flow

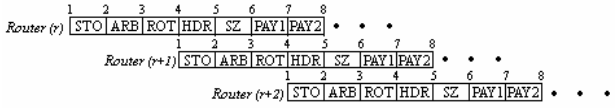


Fig. 2 Transaction model

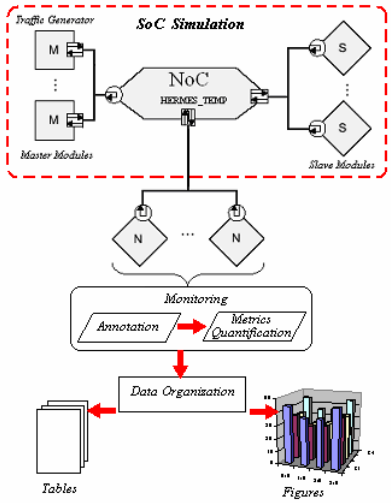


Fig. 3 *NoC* Performance evaluation framework

[1] Created from the Hermes untimed model[8]

the set of 8 configuration parameters shown in Table I. Nested ring is a new topology that we are proposing in this work, see figure 4).

TABLE I. NOC PARAMETERS

| ***Parameter*** | ***Value*** |
|---|---|
| *NoC* size | 4x4 |
| *NoC* type | Homogeneous |
| IP computation component per router | 1 |
| Ports per router | 5 |
| Buffer Size | 4 flits |
| Arbitration mechanism | Round-Robin |
| Topology | Mesh, Torus, Chordal Ring and Nested Ring |
| Routing Algorithm | XY, XY Adaptive, West First, North Last, Chordal, Nested |

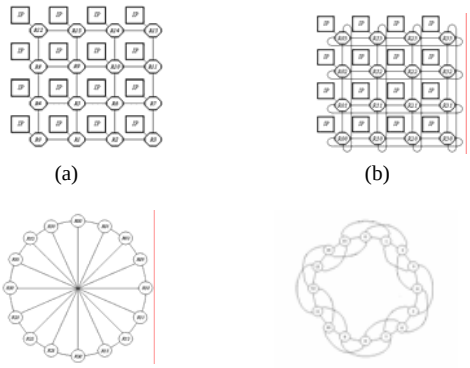


Fig. 4 Topologies (a) Mesh, (b) Torus, (c) Chordal Ring (d)Nested Ring.

### 4.3. Monitors

The monitors annotate the communication events. This information is employed to calculate a set of performance metrics. Our framework calculates six metrics. Two global metrics: 1) average transaction latency (cycles); 2) average transaction throughput (transactions/cycle); and the remaining four are inner metrics: 3) bandwidth (flits/cycle); 4) router average waiting time (cycles); 5) router commutation rate (commutations/cycle); and 6) packet path dispersion (%). The framework presents information about the global and inner *NoC* behaviour/performance.

### 4.4. Framework Validation

With the purpose of evaluating the fidelity of our method, the TLM simulation results were compared with the RTL simulation results. The validation process shows that the design decisions taken at the RTL level would have been the same.

## 5. RESULTS

In order to illustrate the use of our framework we compared 10 different *NoC* instances, each one defined by changing two configuration parameters: topology and routing algorithm (table II). The performance evaluation was based on three traffic patterns: 1) Hot spot (each

master has a preferential slave to communicate); 2) transpose (each *(x,y)* node communicates with its corresponding *(y,x)* node); and 3) pseudo-random (the *NoC* nodes communicate according to the uniform distribution). These patterns were used as *NoC* benchmarks in previous works [6]. The proposed framework allows two types of analyses that allow to rank the different *NoC* instances according to 1) their relative performances; 2) designer defined quality criteria that measure the distance between the estimated *NoC* performance metrics and its optimal value. The quality criteria adopted in this work divides the performance estimation results in three categories: (i) satisfactory *S*, (ii) reasonable *R*, and (iii) deficient *D*. Figure 5a,b show the average transaction latency and average transaction throughput global metrics results. The performance depends of the traffic conditions.

TABLE II. SET OF TESTED CONFIGURATIONS

| Configuration | Topology | Routing |
|---|---|---|
| C1 | *Mesh* | XY |
| C2 | | West First |
| C3 | | XY_Adaptive |
| C4 | | North Last |
| C5 | *Torus* | XY |
| C6 | | West First |
| C7 | | XY_Adaptive |
| C8 | | North Last |
| C9 | *Chordal Ring* | Chordal |
| C10 | *Nested Ring* | Nested |

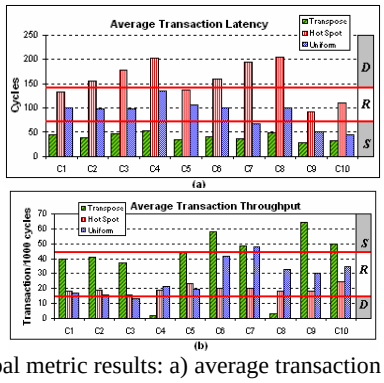


Fig. 5 Global metric results: a) average transaction latency; b) average transaction throughput

Assuming that all three traffic conditions have the same occurrence probability[2] (taking their average), the global metrics lead to the following ranking (Table III) based on the ratio average latency/average throughput and the quality criteria per traffic condition. A new analysis using the inner metrics results (see section 4.3) shows that the discarded C2 and C6 *NoC* instances may present better performance than C1 and C5. An evaluation of the routers average waiting time, show that C2 and C6 present two localized contention points at routers 3 and 15 (Figure 4). Based on this conclusion we incremented the buffers size from 4 to 6 *flits*. Repeating the analysis for all 10 configurations (Table IV).

[2] Alternative traffic condition may be adopted.

TABLE III. NOC CONFIGURATION RANKING (4 FLITS)

| Rank | Configuration | Transaction Latency | Transaction Throughput |
|---|---|---|---|
| 1 | C10 | SRS | SRR |
| 2 | C9 | SRS | SRR |
| 3 | C5 | SRR | RRR |
| 4 | C1 | SRR | RRR |

TABLE IV. NOC CONFIGURATION RANKING (6 FLITS)

| Rank | Configuration | Transaction Latency | Transaction Throughput |
|---|---|---|---|
| 1 | C10 | SSS | SRS |
| 2 | C9 | SRS | SRS |
| 3 | C2 | SRS | SRS |
| 4 | C6 | SRR | SRS |

## 5. CONCLUSIONS

In this work we proposed a framework that allows to estimate the performance of a *NoC* from its high level TLM model. Despite the low precision of such estimations, their fidelity permits a designer to quickly rank different *NoC* instances according to their relative performances for different traffic conditions. We showed how to rank 10 different *NoC* configurations that depended on two parameters: topology and routing algorithm. By adopting average transaction latency and throughput we were able to obtain a first ranking. After including 4 inner performance metrics, we found out that by increasing the input buffer size we could obtain a different ranking and improved performances. Other such analyses may be realized expanding the designer possibility to enhance the *NoC* performance. The early behavior prediction of an optimal *NoC* configuration diminishes the system design time and time-to-market. As a future work, the framework will be used to design application specific *NoCs*.

## 6. ACKNOWLEDGE

This research has been supported by PNM-CNPQ.

## 7. REFERENCES

[1] Benini, L. De Micheli,G. *"Networks on chips: a new SoC paradigm"*. Computer, Volume: 35(1), Jan. 2002, pp. 70-78.

[2] M. Coppola, C. Pistritto, R. Locatelli, A. Scndurra. *"STNoC™: An Evolution towards MPSoC Era"*. NoC Workshop. DATE06.2006.

[3] A. Jalabert, S. Murali, L. Benini, G. De Micheli. *"XpipesCompiler: A tool for instantiating application specific Networks on Chip"*. Proceedings of the Design, Automation and Test in Europe Conference and Exhibition DATE04. 2004.

[4] R. Lemaire, F. Clermidy, Y. Durand, D. Lattard. *"Performance Evaluation of a NoC-Based Design for MC-CDMA Telecommunications using NS-2"*. In IEEE RSP 2005.

[5] P. Pande, C. Grecu, M. Jones, A. Ivanov, R. Saleh. "Evaluation of MP-SoC Interconnect architectures: a Case Study". Proceedings of the 4th workshop on System-on-Chip for Real-time applications IWSOC04.2004

[6] E. Bolotin, I. Cidon, R. Ginosar, A. Kolodny. *"QNoC: QoS architecture and design process for network on chip"*. Journal of system architecture. 2004.

[7] F. Dumitrascu, I. Bacivarov, L. Pieralisi, M. Bonaciu, ª Jerraya. "*Flexible MPSoC Platform with Fast Interconnect Exploration*". DATE06.

[8] F. Moraes, N. Calazans, A. Mello, L. Möller, L. Ost. "*HERMES: an infrastructure for low area overhead packet-switching networks on chip*". In: the VLSI journal 2005.