Simulation and Performance Evaluation of a Network-on-Chip Architecture based on SystemC

Thanh-Vu Le-Van, Xuan-Tu Tran
SIS Laboratory, VNU University of Engineering and Technology, Vietnam National University, Hanoi.
144 Xuan Thuy road, Cau Giay district, Hanoi 10000, Vietnam. Email: vulvt@husc.edu.vn; tutx@vnu.edu.vn

***Abstract*—The Network-on-Chip (NoC) paradigm has been recently known as a competitive on-chip communication solution for large complex systems such as multi-core and/or many-core systems thanks to its advantages. However, one of the main challenging issues for NoC designers is that the network performance should be rapidly and early pre-proved for target applications.**

In this paper, we present a NoC simulation and evaluation platform allowing designers to simulate and evaluate the NoC performance with different network configuration parameters. The proposed platform has been implemented in SystemC to be easily modified to adapt different simulation strategies and to save the simulation time. With this platform, designers can deal with: (i) configuring the network topology, flow control mechanisms and routing algorithm; (ii) configuring various regular and application specific traffic patterns; and (iii) simulating and analyzing the network performance with the assigned traffic patterns in terms of latency and throughput. Obtained results with a 4×4 2D-mesh NoC architecture will be presented and discussed in this paper to demonstrate the proposed platform.

I. Introduction

The on-chip communication mechanism of a conventional System-on-Chip (SoC) is traditionally established using dedicated point-to-point interconnections and shared bus architectures (single bus or hierarchical busses). Nowadays, thanks to the rapid evolution of the semiconductor technology, designers integrate more and more processing elements (i.e., intellectual properties or IPs) into a system to meet the increasingly high demands of target applications. The design of conventional systems basing on shared bus communication architecture therefore encounters many drawbacks such as limited throughput, power consumption, synchronization, etc. [1]. Recently, Network-on-Chip (NoC) paradigm has been proposed and quickly become as a competitive solution for the on-chip communication of large complex SoCs. The advantages of a NoC based system are numerous: high throughput with power efficiency, high scalability and versatility, synchronization [2], [3], [4].

Because of these advantages, many NoC architectures have been designed and developed with different topologies, routing algorithms, end-to-end communication flow control mechanisms, router architectures by research groups at both universities and industries. However, it is important to provide suitable NoC architectures for target applications in order to meet the applications' requirements while keeping a reasonable implementation cost.

The design of NoCs at Register-Transfer-Level (RTL) is time-consuming. It is also very difficult to modify and/or change the network architecture if the design is not suitable for the target application. While high level design is known as a more favorable design methodology providing flexibility and requiring less design time. Therefore, designers can fast model NoC architectures and evaluate the performance to be sure that the design meet the requirements of the target applications in an early design stage. In this way, it seems that SystemC [5], a C++ class library, is a good choice for quickly modeling and simulating NoC architectures. Indeed, because a SystemC model is totally described by a software programming language, the SystemC model can be very flexible and the simulation can run at a faster speed than a RTL model.

In this paper, we first present the targeted NoC architectures, particularly a 4×4 2-D mesh NoC architecture, modeled at system level using SystemC. Then, the network performance in terms of throughput and latency is evaluated by a proposed SystemC based platform which is composed of the NoC architecture and 16 stimulating IP cores. These IPs are developed to generate communication patterns which can be parameterized, and then inject them into the network with predefined application scenarios. The experimental results show the best performance achieved for each type of applications and mapping strategies.

The remaining part of this paper is organized as follows. Section II introduces and discusses related works which have been presented in literature. Section III presents the targeted NoC architectures. Section IV describes the proposed simulation platform with stimulating IP cores, all modeled in SystemC. Network performance evaluation strategies and experimental results are reported and discussed in Section V. Finally, conclusions and future works are given in Section VI.

II. Related Works

Evaluating the performance of a NoC architecture is very important to determine the network parameters to meet the requirements of target applications. This work should be done in earlier design phase to save time as well as to reduce design expenses. To do that, designers have to develop a simulation and performance evaluation platform at high level languages [6], [7], [8]. Sun *et al.* [6] developed a simulation platform for NoC architectures based on NS-2 (a network

978-1-4673-4352-7/12/$31.00 ©2012 IEEE

simulator). As NS-2 is a simulation tool for off-chip communication networks, it is difficult to adapt on-chip network architectures efficiently. In [7], Fen *et al.* used OPNET to simulate NoC architectures with different network topologies as well as commutation modes. In [8], an OMNET++ based simulation platform has been developed for heterogeneous NoCs, called HNOCS (Heterogeneous NoC Simulator). Two kinds of communication traffics have been used to evaluate the network performance: uniform traffic and non-uniform traffic. HNOCS also provides a rich set of statistical measurements at flit and packet levels: end-to-end latency, throughput, transfer latency, etc. However, the simulation and performance evaluation platforms based on high level languages cannot provide accurate results as HDL based plaforms. SystemC is still a good solution for developing simulation platform for NoCs thanks to its flexibility and simulation speed. There are several research groups have developed their own simulation & evaluation platform based on SystemC such as [9], [10], [11]. The work presented in [9] developed a platform allowing to simulate NoC architectures and evaluate their latency and throughput in corresonding with the network load. The other works presented in [10], [11], proposed platforms which can be used to explore NoC architectures for trading-off between the network performance and the power consumption. In our work, we focus on simulating and performance evaluating method and platform which can used for several types of NoC architectures in order to find the suitable network parameters for targeted applications. The following section will introduce the NoC architectures targeted in this work.

III. Targeted NoC Architectures

A. Network topology

There are many topologies can be used for NoC architectures such as fat-tree, butterfly, ring, mesh, torus or folded torus. Within these topologies, 2-D mesh and 2-D torus/folded torus have become dominate thanks to its advantages to be implemented on silicon hardware [4]. In our work, the 2-D mesh topology has been chosed, however, 2-D torus/folded torus topologies are also supported. Figure 1 presents a 4×4 2-D mesh network architecture developed in this work.

The network architecture is composed of network routers, network links, and the interface between the network and the processing elements (called network interface). Each network router has five bi-directional ports which are connected to four neighbouring routers and a nearest processing element. More details of this NoC architecture and its implementation can be found in [12].

B. Routing algorithms and data format

The communication mechanism used in our targeted NoC architectures is packet switching with source, deterministic dimession ordered routing (DOR) algorithms. In particular, the XY, West-First (WF), Nort-Last (NL), and Negative-First (NF) algorithms have been applied for the network in order to avoid deadlock phenomenals. The message is split and encapsulated into packets at the source before being injected into the network. Each packet is composed of a header flit, following by one or more data flits (including body flits and a tail flit). The size of each flit is 34 bits, where 32 bits are used for data and two most significant bits, Begin-of-Packet (BoP) and End-of-Packet (EoP), are used for control purposes. The format of these flits are shown in figure 2.

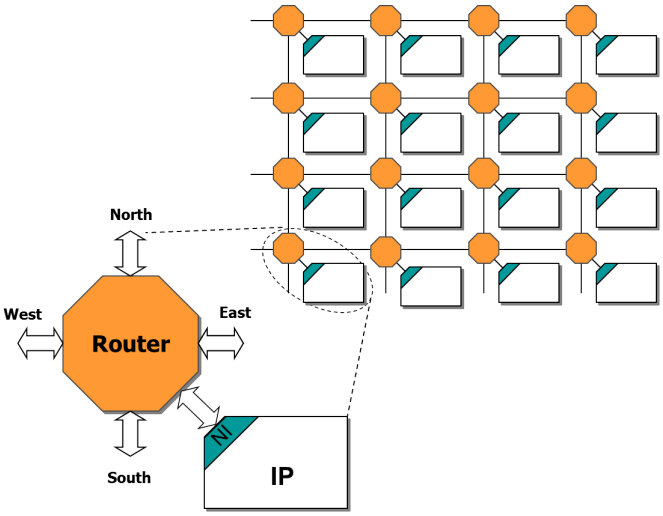

Fig. 1. A 4×4 2-D mesh Network-on-Chip Architecture.

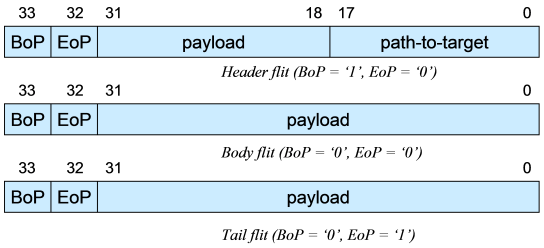

Fig. 2. Flits' format.

Header flit has BoP = 1 and Tail flit has EoP = 1. If both BoP and EoP equal 1, the packet has only one flit while if both values equal 0, this is a body flit. Thanks to these control bits, the network router can easily recognize the type of an incoming flit (header flit, body flit, or tail flit) and make a routing decision. Even using different routing algorithms, the routing arbitrations at the network routers are similar. To route a packet on network, routing information have to be included in the header flit (in the "path-to-target" field). This field will be shifted to the right at each router for next routing direction after two least significant bits are used due to source routing algorithms.

IV. Simulation & Evaluation Platform

As we mentioned above, the NoC architectures should be measured or evaluated for performance before being used in real applications. This ensures the NoC architecture with defined parameters suitable for target applications. In this work, to evaluate the communication performance of NoC

architectures, we proposed a simulation and performance evaluation platform as shown in figure 3. This platform consists of two parts: (i) the network architecture with IP cores, configuration and evaluation methods/strategies setting modules; and (ii) the performance evaluating unit. All these parts are modeled at high level abstraction using SystemC (version 2.2.0) [5], therefore, the platform can be easily modified to adapt different simulation strategies thanks to its flexibility. In addition, as SystemC model can run faster than RTL model, modelling in SystemC allow designers to save the simulation time, especially with a huge amount of data.

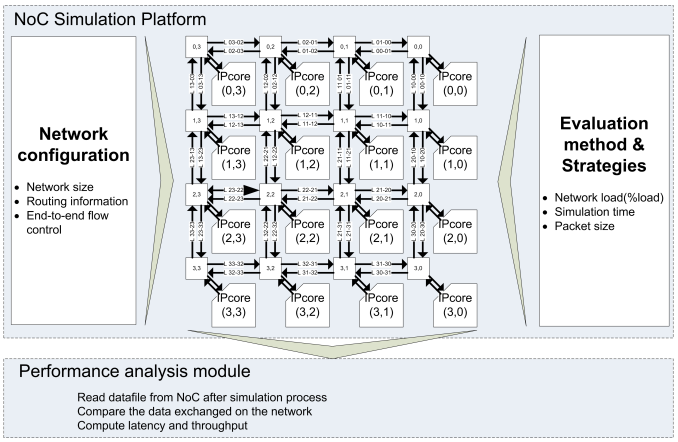

Fig. 3. The proposed platform for NoC performance evaluation.

The first part of the proposed platform allows us to configure the network (network topology, flow control and routing algorithms) as well as set up evaluation methods and strategies (through various simulation scenarios with different network loads, data packet sizes, simulation times). The second part helps us to analyze the network performance in terms of latency and throughput throught the simulation results obtained from the first part with different assigned traffic patterns.

To simulate and evaluate the performance of a NoC architecture, designers just load all the input parameters to configure the NoC as well as the IP cores, and then launch the simulation & evaluation platform. All the information related to the data transfered on the NoC will be saved in ANSI files at all the IP cores involved in the simulation scenario, then will be used by the analysis part to determine the performance of the targeted NoC architecture with the corresponding simulation scenario.

As presented in figure 3, the network architecture has a size of 4×4, however the network size can be extended to $m \times n$ to meet the demand of target applications. The network routers and the attached IP cores are numerated as (x, y) to be addressed by the platform during the simulation, where $0 < x < m$ and $0 <= y <= n$. The results which will be presented in below are obtained from a 4×4 network architecture ($m = n = 4$).

The performance in terms of the latency and throughput of the NoC architecture is evaluated and measured in corresponding to the network load and the size of the packets to be transferred on the network. To do that, the IP cores have to inject data in considering different data loads and sizes of the injected packets. For this target, we have developed an abstraction model of the IP cores as described in figure 4. The IP core has to be able to generate data with different packet size at a balance rate and then inject them to the network. On the other side, the IP core has also be able to receive data from the network. Therefore, each IP core has two sub-blocks: data generating process and data receiving process. The data generating process will generate the data and inject them to the network as configured by the desingers to adapt the simulation scenario. The data receiving process will receive data from the network, process and put them into an ANSI file (to be analyzed by the performance evaluating unit).

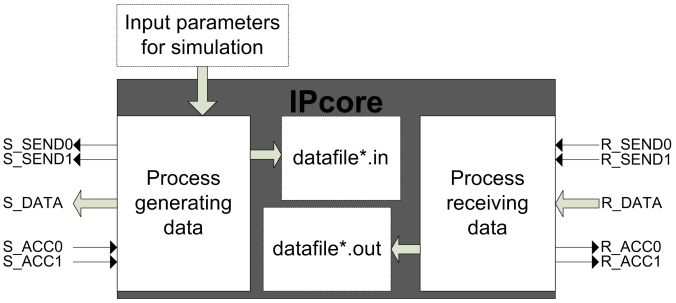

Fig. 4. Simulation IP core.

During simulation, the number of packets are injected into network is proportional to the network load. We propose time balancing mechanism by inserting idle time when number of packets less than saturation. The data injection at each IP core is described in figure 5. At the time when a data flit is injected into the network, it is also stored in ANSI files to be used for evaluating the performance (will be processed by the performance evaluating unit). This technique allows the network operate at maximum performance at each simulation scenario.

Once the simulation finishes, the performance evaluating unit will read all simulation results from ANSI files at IP cores. It compares injecting and receiving data, and calculates the transfering time of all packets. After evaluating for each IP core, it will store the obtained results into ANSI files. The result files contains latency and throughput of the evaluation scenario corressponding to input parameters. More detail of the proposed platform's structure has been presented in [13]. In the next section, we will focus on the method used to evaluate the communication performance of targeted NoC architectures.

V. Performance Evaluation and Experimental Results

The plaform can be configured to simulate and evaluate the targeted NoC architectures with different sizes. However, in this papers only the experimental results obtained from a 4×4 2D mesh topology will be presented.

As mentioned in the previous session, the platform has two parts. The first part is used to configure and lanch the simulation while the second part (the performance evaluating unit) is

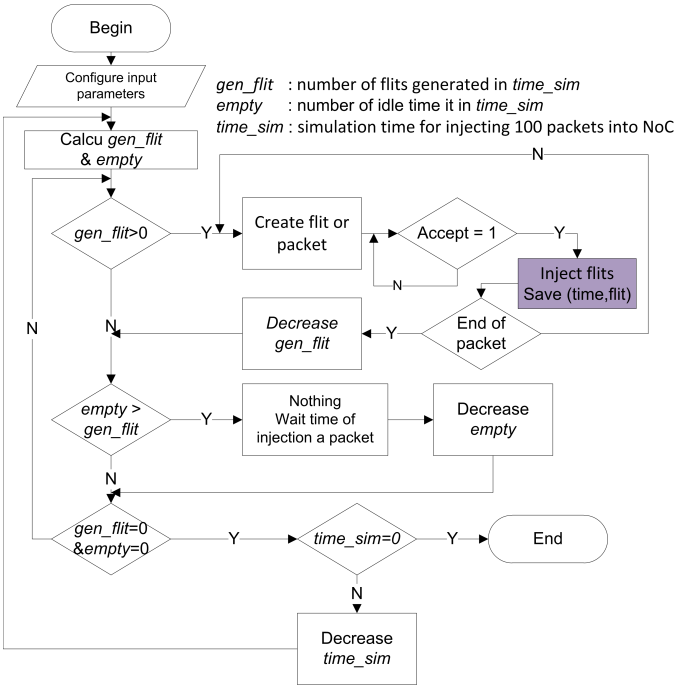

Fig. 5. Flow chart for injecting data at the IP cores.

used to analyze the simulation results to determine the network performance. The performance evaluating unit reads the stored data from ANSI files at the IP cores after the simulation has been terminated. In these ANSI files, we can find neccesary information related to the transmitted data as well as the injecting time and receiving time. Therefore, the performance evaluating unit can easily analyze and calculate the latency of each packet and the total throughput of the network. This unit is also able to compares the content of transferred data between the IPcores to assess the reliability of the target NoC architectures. The platform support many patterns, but in the work we use uniform complement communication pattern as in [14]. All the IP cores inject data into the network with same rate and request communication rescources. A pair sink and destination exchange data through the network combining a closed loop. Example, IP core (0,1) sends data to IPcore (3,2) and IP core (3,2) sends data to IPcore (0,1).

The latency of a packet is measured from the time when the transmitted IP core injects a packet into the network until the destination IP core receives this packet. So, the network latency is the average latency of all packets transmitted in the network [15]. The network latency can be calculated as follows:

$$L = \frac{\sum_i^N L_i}{N} \quad (1)$$

In which, N is number of packets exchanged through the NoC; L_i is the latency of *i*th packet.

The throughput at each network interface represent how many flits through the network interface at the corresponding IP core. In this work, we compute the average throughput, it represent the average value exchanged data over the NoC arcchitecture as mentioned in [15]. The total throughput (TP) will be calculated as follows:

$$TP = \frac{TotalPackets \times PacketSize}{NumberofIPcores + TotalTime} \quad (2)$$

The evaluating strategies for NoC performance have been excuted by changing the input parameters at IP cores such as packet size and injection rate (the rate of injecting data into NoC architecture).

Evaluating scenario 1 For each network configuration set (a defined NoC model), we fix the packet size and increase the injection rate (data load), then repeat the simulation with various packet sizes. The platform will run the simulation and calculate the latency as well as the total throughput of the network. This will help designers to determine the suitable network load for each network configuration with a fixed packet size.

Evaluating scenario 2 For each network configuration set, we fixe the injection load and increase the packet size, then repeate the simulation with various injection rates. The platform will run the simulation and calculate the latency as well as the total throughput of the network. This will help designers to determine the packet size for each network configuration if the network load is already estimated.

To run our simulation and evaluation platform, the following hardware environment and configuration parameters are used.

TABLE I
PARAMETERS FOR SIMULATION

Features	Description
Network configuration	
Topology	4×4 2D mesh
Control flow	Credit based mechanism
Routing algorithm	Source, deterministic XY algorithms (DOR)
Communication pattern	
Traffic pattern	Complement, uniform
Packet size	from 1 to 256 flits
Simulation environment	
HDL	SystemC 2.2.0
Operating system	Linux
Hardware	IBM computer with $2.5GHz$ dual-core Intel processor, $2GBytes$ RAM

Figure 6 shows the performance of the targeted NoC architecture in terms of latency and throughput in correspondings to the network load. The data injected into the network arechitecture have a packet size of $16flits$. From the obtained results, it is clear that the network throughput increasing linearly until the network load reaches to 50%. It means that the communication on the network has good response when the density of transmitting packets is below 50%. Once the traffic is more heavy, the communication will be saturated due to the congestion in arbitrating at the network routers.

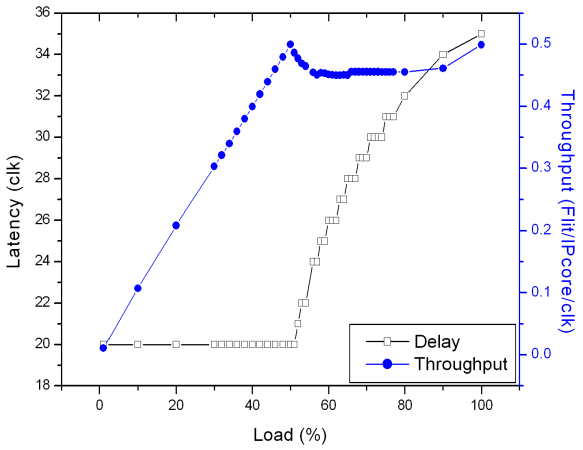

Fig. 6. Evaluation results with 16-flit packets.

In particular, the throughput sligtly decreases ater reaching to 50% before remaing constantly (as discussed in [16]).

Similarly, the latency of the network remains very small, nearly constant, when the network load is less than 50%. The latency will be dramaticly increased when the load is over 50% because there are many packets have to share the limited resources. However, since the targeted NoC architecture is equiped with dead-lock free routing algorithms, then the network still operates when the network load reaches to 100%. In this case, the latency is measured at 35 clock cycles and the network throughput saturates at $0.499 flits/IPcore/clk$.

When we change packet size, the latency of packet increases with the same rule: contant when load less then 50% and dramatically increases when load more than 50% (as shown in figure 7).

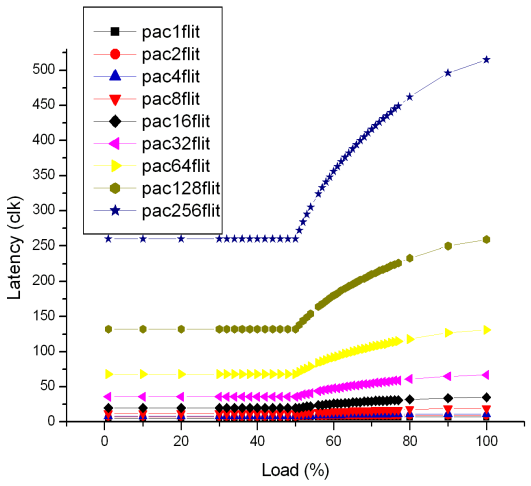

Fig. 7. Network latency in corresponding to the data load (injection rate) with different packet sizes.

Figure 8 shows the average throughput of NoC with different packet sizes. We can see that there are points that the total throughput get the maximum value, at injection rate of 50% and 100%. This is understandable because the packet chaining is best optimal at these injection rates. With 1-flit packets, the

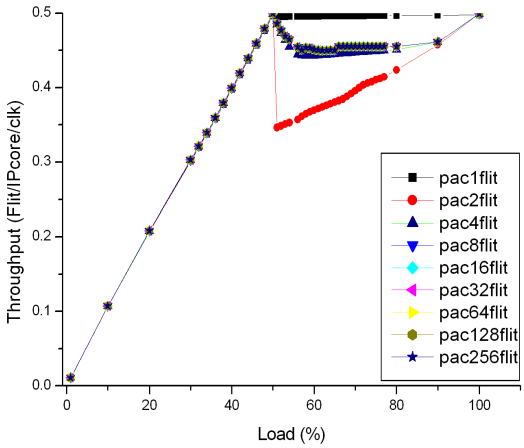

Fig. 8. Total throughput in corresponding to the data load (injection rate) with various packet sizes.

network throughput increases linearly when the network load is less than 50%, then it becomes saturated when the network load over 50%. The network throughput is reduced about 20% with 2-flit packets when the network load over 50% while the other cases reduce 10% as shown in figure 8. The reason that the network throughput decreases suddenly at 50% load is due to the instability of packet chaining (the resource disputing is not optimal at the router when the load is more than 50%).

The obtained results show that the proposed plaform operates stably and accurately as the Register-Transfer-Level (RTL) model with different packet sizes and injection rates while the simulation time is extremly reduced and the configuration can be easily realized thanks to its higher abstraction level.

VI. Summary and Conclusions

Chosing suitable parameters for NoC architectures is an important issue in NoC design and implementation, bringing the NoC paradigm to real applications. In this paper, we have presented a developed simulation and evaluation platform to measure the NoC performance in terms of network latency and throughput. The proposed platform has been designed and implemented in SystemC, a high level abstraction language, allowing designers to shorten the simulation time in chosing NoC parameters to meet the requirements of the target applications as well as to be easilly modified by the users. With this platform, designers can configure the network topology, flow control mechanisms and routing algorithm as well as configure a various regular and application specific traffic patterns. The platform allows designers to analyze automatically the network performance with the different traffic patterns and data load. Experimental results with a 4×4 2D-mesh NoC architecture have been reported and discussed.

ACKNOWLEDGEMENTS

This work is partly supported by Vietnam National University, Hanoi (VNU) through research project No. QGDA.10.02 (VENGME).

REFERENCES

[1] W.J. Dally and B. Towles. Route Packets, Not Wires: On-Chip Interconnection Networks. In *Proceedings of the Design Automation Conference (DAC)*, pages 684–689, Las Vegas, June 2001.

[2] L. Benini and G. De Micheli. Networks on Chips: A New SoC Paradigm. *IEEE Computer*, 35(1):70–78, January 2002.

[3] Pierre Guerrier and Alain Greiner. A Generic Architecture for On-Chip Packet-Switched Interconnections. In *Proceedings of Design, Automation and Test in Europe Conference and Exhibition(DATE)*, pages 250–256, Paris, 2000.

[4] Edith Beigné, Fabien Clermidy, Pascal Vivet, A Clouard, and M. Renaudin. An Asynchronous NoC Architecture Providing Low Latency Service and its Multi-level Design Framework. In *Proceedings of the ASYNC'05*, pages 54–63, New York, March 2005.

[5] Open SystemC Initiative (OSCI). *IEEE Standard SystemC Language Reference Manual*.

[6] Yi-Ran Sun, Shashi Kumar, and Axel Jantsch. Simulation and Evaluation for a Network on Chip Architecture Using Ns-2. In *Proceedings of the 20th NORCHIP conference*, 2002.

[7] Ge Fen, Wu Ning, and Wang Qi. Simulation and Performance Evaluation for Network on Chip Design using OPNET. In *Proceedings of TENCON*, pages 1–4, November 2007.

[8] Yaniv Ben-Itzhak, Eitan Zahavi, Israel Cidon, and Avinoam Kolodny. HNOCS: Modular Open-Source Simulator for Heterogeneous NoCs. In *Proceedings of the 2012 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, July 2012.

[9] Chai Song, Chang Wu, Yubai Li, and Zhongming Yang. A NoC Simulation and Verification Platform Based on SystemC. In *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, pages 423–426, 2008.

[10] Basavaraj Talwar and Bharadwaj Amrutur. A SystemC based Microarchitectural Exploration Framework for Latency, Power and Performance Trade-offs of On-Chip Interconnection Networks. In *Workshop on NOC Architectures*, Lake Como, Italy, November 2008.

[11] Stefano Gigli and Massimo Conti. A SystemC Platform for Network-on-Chip Performance/Power Evaluation and Comparison. In *Proceedings of the 7th Workshop on Intelligent Solutions in Embedded Systems*, pages 63–69, 2009.

[12] Nam-Khanh Dang, Thanh-Vu Le-Van, and Xuan-Tu Tran. FPGA Implementation of a Low Latency and High Throughput Network-on-Chip Router Architecture. In *Proceedings of the 2011 International Conference on Integrated Circuits and Devices in Vietnam (ICDV)*, pages 112–116, Hanoi, Vietnam, August 2011. IEICE.

[13] Thanh-Vu Le-Van, Dien-Tap Ngo, and Xuan-Tu Tran. A SystemC based Simulation Platform for Network-on-Chip Architectures. In *Proceedings of the 2012 IEICE International Conference on Integrated Circuits and Devices in Vietnam*, pages 132–136, Danang, Vietnam, August 2012.

[14] Lionel Ni Jose Duato, Sudhakar Yalamanchili. *Interconnection Networks: An Engineering Approach*. Morgan Kaufmann Publishers, 2003.

[15] Partha Pratim Pande, C. Grecu, M. Jones, A. Ivanov, and R. Saleh. Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures. *IEEE Transactions on Computers*, 54(8):1025–1040, aug. 2005.

[16] George Michelogiannakis, Nan Jiang, Daniel Becker, and William J. Dally. Packet Chaining: Effcient Single-Cycle Allocation for On-Chip Networks. *IEEE Computer Architecture Letters*, 10(2):33–36, 2011.