PROCEEDINGS

55. IWK

Internationales Wissenschaftliches Kolloquium
International Scientific Colloquium

13 - 17 September 2010

Crossing Borders within the ABC

Automation,

Biomedical Engineering and

Computer Science

Faculty of
Computer Science and Automation

www.tu-ilmenau.de

TECHNISCHE UNIVERSITÄT
ILMENAU

Impressum
Published by

Publisher:	Rector of the Ilmenau University of Technology Univ.-Prof. Dr. rer. nat. habil. Dr. h. c. Prof. h. c. Peter Scharff
Editor:	Marketing Department (Phone: +49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de) Faculty of Computer Science and Automation (Phone: +49 3677 69-2860) Univ.-Prof. Dr.-Ing. habil. Jens Haueisen
Editorial Deadline:	20. August 2010
Implementation:	Ilmenau University of Technology Felix Böckelmann Philipp Schmidt

USB-Flash-Version.

Publishing House:	Verlag ISLE, Betriebsstätte des ISLE e.V. Werner-von-Siemens-Str. 16 98693 Ilmenau
Production:	CDA Datenträger Albrechts GmbH, 98529 Suhl/Albrechts
Order trough:	Marketing Department (+49 3677 69-2520) Andrea Schneider (conferences@tu-ilmenau.de)

ISBN: 978-3-938843-53-6 (USB-Flash Version)

Online-Version:

Publisher:	Universitätsbibliothek Ilmenau ilmedia Postfach 10 05 65 98684 Ilmenau

Home / Index:
http://www.db-thueringen.de/servlets/DocumentServlet?id=16739

SURVEY OF NETWORK-ON-CHIP RESEARCH AND MODELING APPROACHES

Andriy Osadchuk, Wolfgang Fengler

Institute of Computer Engineering, TU Ilmenau, Ilmenau, Germany
{andriy.osadchuk, wolfgang.fengler}@tu-ilmenau.de

ABSTRACT

In future embedded systems Networks-on-Chip (NoC) are expected to play a key role. Especially in multiprocessor systems, where multiple heterogeneous components have to be connected, efficient communication is a very important issue. The main topic of this paper is to give a research survey over the last years in the field of networks-on-chip. We consider basic guidelines and advantages of NoC research. As a subset of System-On-Chip (SoC), NoCs are an integral part of SoC design and like in every design flow the modeling and simulation are important. It is necessary to model the behavior and to realize virtual prototypes of such systems in order to evaluate the advantages of their technological implementation in early design stages. Several modeling techniques that are used in NoC design are also considered.

Index Terms – System-on-Chip, Networks-on-Chip, Networks-on-Chip modeling.

1. INTRODUCTION

The growing integration density of today's technologies makes the number of transistors per chip area increase more and more. This fact enabled the increase of processing resources on a single chip. Such Systems-on-Chip (SoC) consist of complex assemblies, such as processors, accelerators for multimedia and encryption applications, memory, communications and interface controllers. Modern SoC have high demands on the underlying communication infrastructure. To connect groups of elements on the chip in SoC basically SoC–bus were used such like AMBA by ARM, WishBone by Silicore and STBUS by Microelectronics etc.[1]. Traditional approaches, such as shared buses or circuit-switched buses have only limited abilities. The main disadvantage is their centralized layout and that leads to significant link delays. The efforts to decentralize these bus systems resulted in the idea to use the common principles of computer networks. In this case Networks-on-Chip are meant. This approach was first introduced by W.J. Dally and B. Towles in 2001 [2]. The main idea was to replace the global on-chip wires with on-chip interconnection networks to overcome physical limitations of bus based architectures. Furthermore, this architecture was used to reach better reusability and high-performance of the system.

The paper is organized as follows: Chapter 2 sketches Networks-on-Chip and their basics. In Chapter 3, the modeling of NoC and some modeling approaches are described. Conclusions of this paper are given in Chapter 4.

2. NETWORKS-ON-CHIP

In this chapter we describe Networks-on-Chip architectures and their basics. A few examples of NoCs are also presented here.

The NoC development was mainly affected by internet, computer networks and distributed systems. The principles of normal computer networks have been transferred to the SoC level. These principles are: packetized communication, flexible switched networks, abstraction of the communication, protocol hierarchy, fault-tolerance and Quality-of-Service. In this case we can say that Networks-on-Chip are conform to the ISO OSI model. For better understanding, in [3] a comparison between ISO OSI model and NoC research areas was presented.

By the NoCs design the communication and computing are separated, because a Network-on–Chip on itself provides the protocols and interfaces, which separate the communication from implemented services and functions. This separation gives certain flexibility in the design. The design of NoC can be partitioned in 2 phases. In the first one there will be defined topology and size of the network, number of switches, form and type of resources. So we can say that the architecture for the NoC is designed independently of the functionality of future SoC. In the second phase the whole system will be implemented and the appropriate tasks, algorithms or IP-Cores on the designed architecture will be mapped.

2.1. NoC basics.

In the research there are a lot of different approaches where NoCs are used, for instance in [4,5]. Typically these are packet-switched networks that have three fundamental blocks, namely, *switches* (also denoted

as *router*), *network interfaces* (NI) and *links*. The instantiation of the NoC relies on deploying a set of these components to form a topology and by configuring them.

2.1.1. Topology

Topology concerns the layout and connectivity of the nodes and links on the chip. By the choice of the network topology, there are different variants possible. Like analyzed in [6,7] the most used topology is the mesh-topology (Fig.1), because of its optimal characteristics such like optimal wiring (links have limited length) and relatively simple routing. Of course other topologies like ring, torus or tree etc. can be used, some of them are depicted in the Fig.2

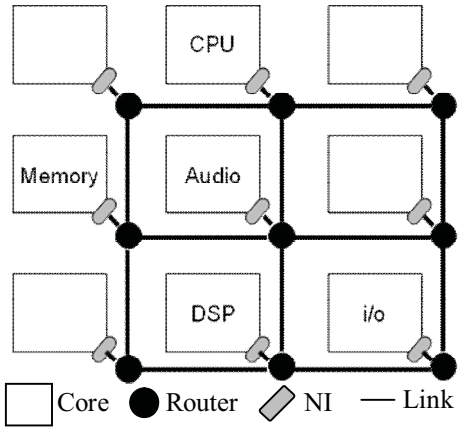

Figure 1 An example of 3x3 mesh-NoC

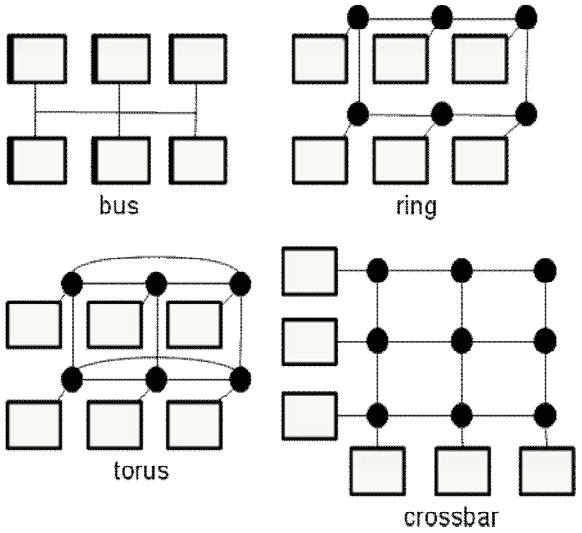

Figure 2 Examples of possible topologies

2.1.2. Switching policy

Once the topology of a NoC has been selected, the switching policy must be determined. The switching policy defines how data through the routers flow. A special feature by the data-transfer in NoCs is that the actual data will be split into flits (*flow control units*). These flits constitute the data-packets.

Two main types of switching are used: circuit switching and packet switching. Circuit switching (Fig.3) reserves a dedicated end-to-end path from the source to the destination before starting to transmit the data. The path reservation is released after transmission of the message. In packet switching the message will be split into a sequence of packets. Each of the packets typically contains header, payload and tail. In header the routing information is represented. The payload contains the actual data and tail contains the end of the packet.

There are three types of packet switching: *wormhole*, *store and forward*, and *virtual cut through*.

***-Store and forward* (SAF)** is the simplest form of the packet switching. The complete packet has to be stored by the node before it is forwarded to the next node. Thus, there is no need for dividing a packet into flits. The main disadvantage of this switching scheme is the relatively big buffer size, because it has to be equal to the size of the packet.

***-Virtual cut through* (VCT)** forwards the parts of the packet when the header information is available. In contrast to the SAF, there is no need to wait for the receiving of the complete packet. Other flits follow the header flit. However, the buffer requirements are equal to the SAF switching [3].

-Wormhole switching is the most used switching technique [6,9]. It is similar to the VCT scheme. The packet is also split into the flits but the difference is that the buffer size of the switch is reduced to one flit, so that there is no need to hold complete packet. The data flits follow the same path as the headers flit. The major disadvantage is that, when a packet is blocked, it blocks the links along the entire route of the packet.

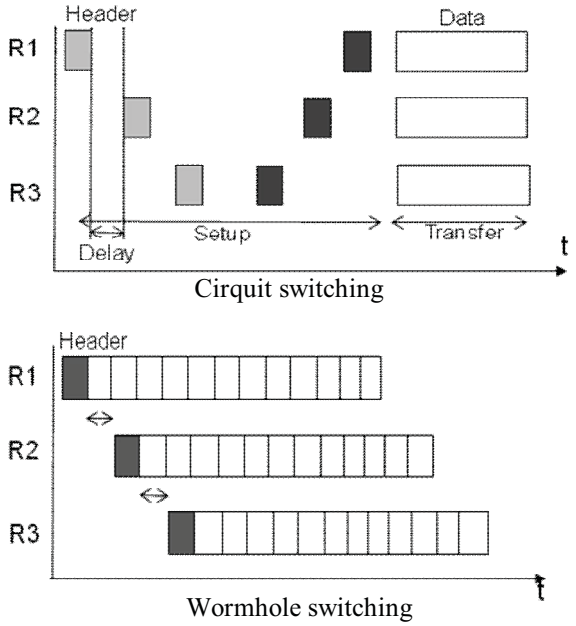

Figure 3 Switching schemas

2.1.3. Routing

The routing determines the routing paths that the packets may follow through the network. In terms of path diversity and adaptivity, routing schemes can be classified into three categories: deterministic routing, oblivious routing and adaptive routing.

In the deterministic schemes all packets choose always the same path from the source node to the destination node. Oblivious routing, which includes deterministic algorithms as a subset, considers all possible multiple paths from the source node to the destination node.

The third category is the adaptive routing. The packets are distributed dynamically according to the network state. The network state may include the status of a node or link, the length of queues, and network load information. Consequently, the route of the data may vary.

The well known and most used routing techniques are XY-Routing (also called dimension routing) [10], and deflection routing. More detailed description of different routing schemes can be found in [1,11].

2.1.4. Quality-of-Service

NoC design requires to adapt a set of parameters. These parameters need to be integrated with the Quality-of-Service(QoS) requirements for designing an embedded system. QoS refers to the services for guarantees of data transfer. QoS parameters and its definition are extremely diverse and application-specific in embedded systems. QoS in NoC based system may be expressed in latency, throughput or jitter of traffic flows between nodes that communicate.

Basically two categories of services are suggested: Best Effort (BE) and Guaranteed Throughput (GT). GT often requires predictability of the traffic load conditions, a feature which often desirable in real-time systems. And the BE category improves the average resource utilization, but no guarantees are given for throughput or latency in general case. Most of QoS issues are coupled with routing and flow control policies [8,12,13].

3. NOC MODELING

The Networks-on-Chip design is covering several abstraction levels, ranging from the transaction to the physical levels. In recent years reuse has been exploited to enhance system design productivity, whereby a design is the result of combining predesigned and preverified components. Because of the complexity to connect heterogeneous elements, the on-chip communication issues are becoming more and more important. Modeling, especially at system level, has become a fundamental step on building SoC/NoC, because it reduces the costs of product development [14]. In the research literature various approaches to NoC modeling are proposed. Some examples can be found in [15-18]

The NoC models can be either analytical or simulation based, or also emulation on FPGA-based and can model communication across abstractions. Every of these approaches have its advantages and disadvantages. On the one hand, simulation based approaches provide highly detailed and accurate results, but on the other hand they need long simulation times. For large NoCs, reducing of simulation details can be required to achieve reasonable simulation times. In contrast to simulation based, the emulated on FPGAs approaches reduce simulation time significantly. However, the emulation on the FPGA has three key drawbacks:
(i) any change in the simulated NoC requires manual redesign of the emulator HDL, (ii) redesign in turn requires complete compilation/synthesis of the FPGA, and (iii) the maximum simulatable NoC size is determined by the available FPGA capacity.

In the following, some modeling approaches will be introduced. We give a short description of key points of approaches represented here.

3.1. SystemC-based approach

Like we analyzed, for the design and modeling of SoC/NoC systems, the modeling language SystemC is wide used. SystemC is a free, C++ based modeling language proposed by the Open SystemC Initiative which spans from concept to implementation in hardware and software [19]. Actually, it is a class library that extends C++ language with new data types, macros and functions. The system components of different abstraction levels, design domains and applications are combined in one system model.

In [20] one on SystemC based, a NoC simulation and verification platform (NSVP) was proposed. The NoC node was modeled and realized in form of a general modularized NoC node structure. The hardware blocks are modeled as logical functional modules that communicate through ports. These ports are connected to the network nodes which can with each other communicate in different ways. For the building of network architecture, two network topologies are used: mesh and torus, which sizes may vary. Routing decisions are made using the XY-routing algorithm. Finally, the designed system can be simulated using the SystemC simulation kernel. The infrastructure of NSVP platform is given in Fig.4.

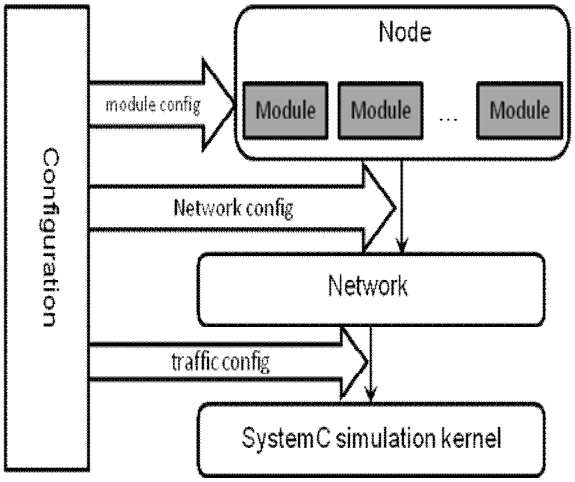

Figure 4 Infrastructure of NSVP

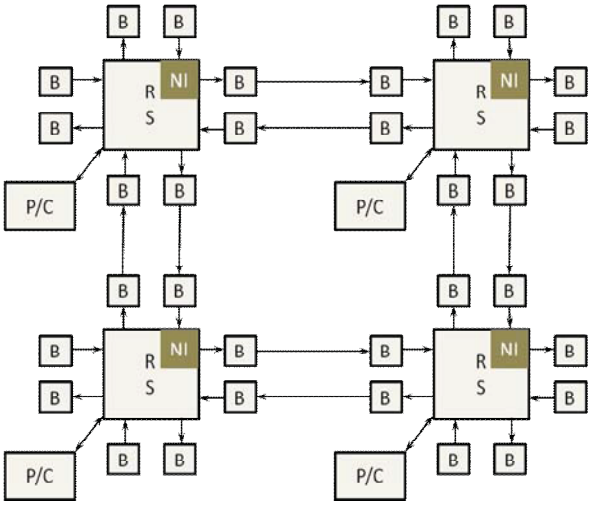

Figure 5 Modeled 2x2 mesh NoC

3.2. Concurrency modeling in MLDesigner

Another system modeling approach is presented in [21]. Authors developed a high-level concurrency model for a multiprocessor system. This model was described in [22]. For the modeling of underlying interconnection, the MLDesigner was used. MLDesigner represents a system-level design and modeling environment that allows modeling of a system at the abstraction levels. Modeling in different domains such as the Discrete Event (DE), Synchronous Data Flow (SDF), Finite State Machines (FSM), Dynamic Data Flow (DDF) domains, is supported there. A combining of multiple domains can further represent the system model.

All NoC building blocks, also called *classes*, were modeled in MLDesigner. These classes were identified from a high level system specification. These classes are: *Producer* (P), *Consumer* (C), *Input Buffer* (IB), *Output Buffer* (OB), *Scheduler* (S) and *Router* (R). The *Producer* class defines a resource and a resource network interface (RNI). The *Input Buffer* class corresponds to an input buffer, a buffer scheduler, virtual channels and a virtual channel allocator. The *Router* class corresponds to a router; and the *Scheduler* class corresponds to a switch allocator. Different parameters were used to model these components. For instance, varying buffer size for in- and output buffers, diverse scheduling criteria for schedulers and buffers, data packet priorities (High, Middle and Low), and routing algorithms (dimension routing algorithms like X-first or Y-First, XY-random). In Fig.5 is shown an example of the modeled architecture. B, P, C, R and S represent buffer, producer, consumer, router and scheduler respectively.

The feasibility of this modeling approach was proofed with mapping a H.264 decoder onto 4x3 mesh-based NoC architecture [21].

3.3. Performance modeling with Deterministic and Stochastic Petri-Nets

One interesting approach for modeling of Networks-on-Chip is proposed in [23]. Authors performed a performance modeling in order to find suitable communication architecture for NoC. Deterministic and Stochastic Petri-Nets (DSPNs) have been used to model on chip communication schemes. Here authors have extended DSPNs to model NoC architectures with different communication modules and processor cores. Using different communication structures, namely Avalon (by Altera) and a more complex custom-made hierarchical communication infrastructure, the predictions of the modeling was verified against actual implementation on a FPGA testbed. For these purposes an FPGA-based platform has been utilized on which several proprietary so-called soft-core processors (Nios and Nios II) besides other components like DMA-controllers, on-chip memories or logic blocks, instantiated and connected. It could be shown that the modeling results are very close to the values measured on an FPGA testbed. Additionally one generic DSPN model based on the regular mesh topology with deterministic routing has been built and analyzed. Further details and detailed description of modeling with Deterministic and Stochastic Petri Nets can be found in [23].

4. CONCLUSION

Networks-on-Chip have emerged as a promising structured way of realizing interconnections on silicon, overcoming the limitations of bus-based solutions. In this paper an overview was given on current approaches in the network-on-chip research

area. We analyzed and presented an average set of currently utilized properties. Also we considered modeling aspects of NoCs. But still, a standard approach to model on-chip interconnection is needed. Several modeling approaches were considered here that, in our opinion, have the potential to be denoted as standard modeling techniques.

5. REFERENCES

[1] G. de Micheli, L. Benini, "Networks on Chips: Technology and Tools", Morgan Kaufmann, San Francisco, 2-41,148-195, 2006.

[2] W. Dally and B. Towles, "Route Packets, Not Wires: On-Chip Interconnection networks", *Proceedings of the 38th Design Automation Conference,* Nevada, USA. 2001.

[3] T. BJERREGAARD AND S. MAHADEVAN, "A Survey of Research and Practices of Network-on-Chip, *ACM Computing Surveys*, Vol. 38, March 2006.

[4] K. Goossens, J. Dielissen, and A. Radulescu, "Aetheral Network on Chip: Concepts, Architectures, and Implementations", *IEEE Design and Test of Computers,* 2005.

[5] M. Milberg, E. Nilsson, R. Thid, Shashi Kumar, and A. Jantsch. The Nostrum backbone – a communication protocol stack for networks on chip. *VLSI Design Conference,* Mumbai, India, January 2004.

[6] E. Salminen, A. Kulmala, and T. D. Hämäläinen, "On network-on-chip comparison", *10th Euromicro Conference on Digital System Design Architectures, Methods and Tools*. 2007.

[7] C. Bobda, A. Ahmadinia. "Dynamic Interconnection of Reconfigurable Modules on Reconfigurable Devices," *IEEE Design and Test of Computers*, vol. 22, no.5. 2005.

[8] A. Agarwal, C. Iskander, R. Shankar, "Survey of Network on Chip (NoC) Architectures & Contributions", *Journal of Engineering, Computing and Architecture* Vol. 3, Issue 1, 2009.

[9] Erno Salminen and Ari Kulmala. and Timo D. Hämäläinen, "Survey of Network-on-chip Proposals". WHITE PAPER, (March),1–13, 2008.

[10] C. Bobda, A. Ahmadinia, M. Majer, J. Teich, S. Fekete, J. van der Veen "DyNoC: A Dynamic Infrastructure for Communication in Dynamically Reconfigurable Devices" *Field Programmable Logic and Applications. International Conference*, 24-26 Aug. 2005.

[11] A. Lu, M. Zhong, and A. Jantsch. "Evaluation of On-Chip Networks Using Deflection Routing". *Great Lakes Symposium on VLSI (GLSVLSI)*, Philadelphia, USA, 2006.

[12] M. A. Al Farouque, "Runtime Adaptive System-on-Chip Communication Architecture", Ph.D. Thesis, Fakultät für Informatik der Universität Fridericiana zu Karlsruhe (TH), 11-12, 2009.

[13] E. Bolotin, I. Cidon, R.Ginosar and A. Kolodny "QNoC: QoS Architecture and design process for Network on Chip", *Journal of Systems Architecture: EUROMICRO Journal*, Vol. 50, 2004.

[14] A. Gerstlauer, D. Shin, R. Dömer. Daniel D. Gajski. "System-level communication modeling for network-on-chip synthesis", IEEE, *Asia and South Pacific Design Automation Conference,* 2005.

[15] N. Genko, D. Atienza, G. De Micheli, NoC "Emulation on FPGA: HW/SW Synergy for NoC Features Exploration". *International Conference ParCo.,*2005.

[16] J. Schmaltz and D. Borrione, "A generic Network on Chip Model.", *Proc. TPHOLs'05*, August 2005.

[17] A. Kohler, M. Radetzki, Modellierung und Simulation von Networks-on-Chip mit OSCI TLM2. *12. Workshop Modellierung und Verifikation von Schaltungen und Systemen*, Berlin, März 2009.

[18] Zhonghai Lu, Rikard Thid, Mikael Millberg, Erland Nisson, and Axel Jantsch. „NNSE: Nostrum network-on-chip simulation environment". *Swedish System-on-Chip Conference (SSoC'03)*, April 2005.

[19] www.systemc.org, Open SystemC Initiative, 2010

[20] Song Chai, Chang Wu, Yubai Li, Zhongming Yang, "A NoC Simulation & Verification Platform based on SystemC", *IEEE International Conference on Computer Science and Software Engineering*. 2008.

[21] A. Agarwal, C.-D. Iskander, H. Kalva, and R. Shankar, "System-level modeling of a NoC-based H.264 decoder", *IEEE Systems Conf.*, Montréal, April 2008.

[22] A. Agarwal and R. Shankar, "A concurrency model for NOC design methodology", *IEEE Conf. on High Performance Computing*, MIT, Sept. 2006.

[23] H.Blume, T. von Sydow, D. Becker, T.G. Noll, "Application of Deterministic and Stochastic Petri Nets for Performance Modelling of NoC Architectures", *Journal of Systems Architecture*, Vol. 53, Issue 8, pp. 466-476, 2007.