E-Core – A Configurable IP Core for Application-specific NoC Performance Evaluation

Stephan Kubisch, Harald Widiger, Claas Cornelius, Dirk Timmermann, Andy Strzeletz
University of Rostock, Institute of Applied Microelectronics and Computer Engineering
18051 Rostock, Germany, Tel.: ++49 (0)381 498 - 7271
{stephan.kubisch,harald.widiger,claas.cornelius,dirk.timmermann}@uni-rostock.de

I. Introduction

During the last years, Networks-on-Chip (NoCs) have become a true alternative for the design of complex integrated Systems-on-Chip (SoCs). Although NoCs are widely used in ASIC design for complex and multi-processor SoCs, we mainly address NoCs on FPGAs where aspects like increasing design complexity, parasitics, and end-to-end latency have to be considered similarly. However, since NoCs are not yet matured in current design flows, EDA tools, and industrial applications, a lot of research is still necessary. Thus, one hot research area is the simulation and evaluation of NoCs in general and with regard to feasibility for a given application scenario [1]. This poster presents an evaluation IP core called E-Core, which allows functional simulation and performance evaluation of NoC architectures. E-Core is implemented in VHDL on the register-transfer-level (RTL). It is derived from our high-level simulation environment previously presented in [2], [3]. Simulation and verification is feasible at every step in the design process: using abstract models in software for preliminary design space exploration, functional verification with common simulation tools, and evaluation in programmable hardware, which immensely speeds up simulation time, e.g., using FPGA prototyping boards.

Section II describes some related work. Section III briefly presents the evaluation IP core. The paper concludes in Section IV.

II. Related Work

An event-based C++ simulator was presented in [4]. It operates on a high abstraction level and is feasible for preliminary design explorations for different application scenarios, e.g., in the networking and telecommunication area. A more complex simulation environment was published in [5]. It serves many scenarios from general multi-processor-designs to application-specific designs and allows for bit- and cycle-accurate simulation. Since it is relatively sophisticated, the simulator is designed for software simulation although the authors envisage a silicon implementation in the future. But this will result in expensive ASIC prototypes. In [6] and [7], SystemC simulation environments for NoCs were presented. Both mainly address analysis of power consumption and design space exploration for different NoC topologies and their side-effects on the performance for a given application. High-level SystemC mechanisms are used to model the IP cores and NoC components on the transaction level. In [8], an emulation framework on an FPGA development board was described. The FPGA's hardwired PowerPC is used for simulation orchestration within an ×pipes-type NoC [9]. A serial connection to a workstation is used for monitoring and analysis.

III. The Evaluation IP Core

E-Core is a traffic generation and monitoring module, which emulates the typical behavior of processing units and IP cores attached to the routers in NoC communication grids. It models the behavior of processing units at the boundaries of their interfaces to the NoC routers. NoCs are known for a strict separation of communication and computation. E-Core also conforms to this principle. Different NoC topologies as well as globally asynchronous locally synchronous (GALS) structures can be evaluated since E-Core's operation is independent from the NoC infrastructure.

E-Core does *not* emulate non-deterministic, reactive behavior or microprocessor functions as proposed in [5] for example. As our own research addresses packet processing, E-Core is primarily applicable for suchlike applications as introduced in [10]. Furthermore, E-Core's functional spectrum is qualified for different types of streaming media scenarios like cryptography, graphic processing, and filter functions. IP cores and modules within packet processing SoCs show various typical patterns of behavior. The list below contains these types. E-Core can be configured to emulate each.

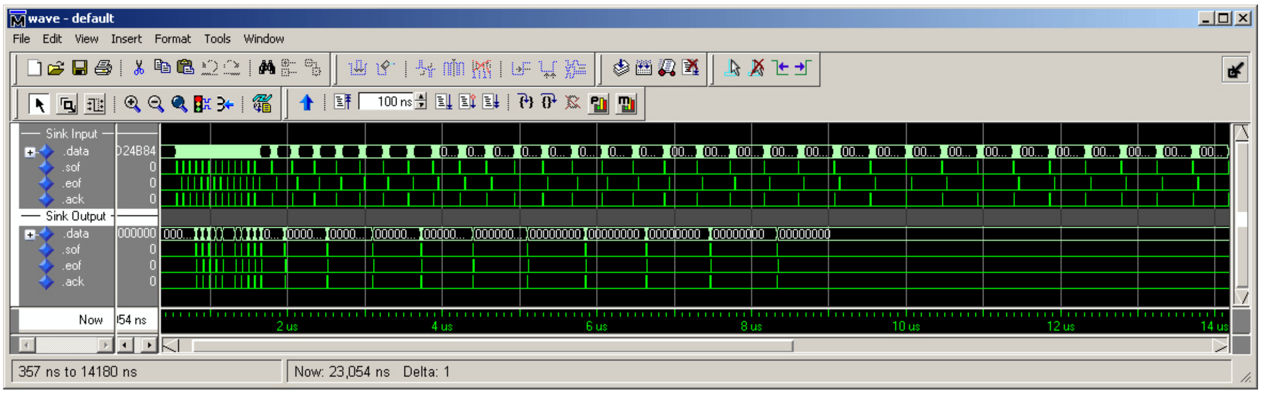
A. Configuration Types

Source As a traffic source, E-Core acts as a pure traffic generator. The injection rate is configurable as well as NoC-specific parameters suchlike flit size, maximum and minimum packet size and packet size distribution, and target addresses.

Sink As a data sink, E-Core only consumes packets at a configurable acceptance rate.

Source & Sink A more sophisticated type is a combination of data sink and source. This configuration is used to emulate a communication between IP cores. I/O ratio, packet sizes, wait cycles, delays, and target addresses are configurable.

Processing element Similar to the sink & source case is the configuration as a typical packet processing element. Here, arriving packets are forwarded to a fixed destination.



357 ns to 14180 ns

Now: 23,054 ns Delta: 1

Fig. 1. ModelSim Waveform with ICMP Source Quenching Mechanism

The delay generated by the processing element is configurable (static or dynamic) as well as the I/O ratio. Within suchlike modules, processed packets may be modified and thus extended or truncated in length. This behavior is configurable with relative and absolute parameters.

General parameters Additionally, each instance of E-Core needs to be configured with its own address and an opcode, which represents one of the configurations mentioned above. Depending on the opcode, only the necessary VHDL state-machines are synthesized to keep a (preferably) low hardware footprint. For the generation of pseudo-random numbers, payload and other randomized parameters, linear feedback shift registers are used, since they can easily be implemented in FPGAs.

Statistics Statistics are minuted within each instance of E-Core. This are, e.g., the numbers of bytes, flits, and packets, which have been sent, received, and had to be discarded. Therefrom, conclusions on each module's and the whole system's throughput and performance can be made for a certain topology and IP core mapping.

B. *Online Capabilitiess*

The core is implemented in RTL-VHDL and thus cycle-accurate. When synthesized and mapped onto a target FPGA, online simulation runs can be carried out in a fraction of the time, which is required with common software simulators. By making use of the Xilinx ChipScope Pro debugging tools, statistics can be gathered directly from the hardware prototype.

Furthermore, features known from the Internet Control Message Protocol (ICMP) are used to self-adjust a source's packet injection rate to a sink's maximum acceptance rate. This is automatically done during operation. The simple ICMP source quenching mechanism is used within each E-Core instance, which is configured as data sink or processing element. When too much packets arrive and cannot be received or processed, a source quench message is sent back to the packets' origin. At the source, the injection rate is decreased by a small, specific fraction with each incoming source quench message.

This behavior is shown in the Modelsim waveform plot in Figure 1. The sink's input signals (Sink Input) show a coarse overview of the arriving packets. These packets arrive with a certain data rate. The sink is not able to consume and process the packets at this high data rate. Thus, it generates short source quenching packets, which is indicated by the activity on the sink's output signals (Sink Output). When these control messages arrive at the source, it decreases its packet injection rate (at circa 2µs), which can be derived from the increasing idle cycles between the packets arriving at the sink's input. After a while (at circa 8.5µs), the sink stops the generation of source quenches since it can accept every incoming packet without any packet loss.

In real networks, ICMP opens up various scenarios for connection-reset, throughput-reduction, and performance-degrading attacks against the Transport Control Protocol (TCP) and other protocols [11] because ICMP itself does not specify any validation checks on incoming control messages. Thus, some ICMP features are deprecated and not used in real networks [12]. But for the evaluation of NoC-based SoCs as described here, suchlike security issues are not of importance.

IV. Conclusion

This poster presents E-Core—a flexibly configurable IP core for the simulation and evaluation of NoC-based systems. It is especially suited for emulating packet processing tasks as well as for similar streaming traffic scenarios. Since E-Core is synthesizable and can be implemented on an FPGA prototyping board, it radically reduces simulation time with online simulations runs in hardware. Future work comprises the automatic transmission of statistics via Ethernet MAC to a connected workstation to further ease the evaluation. In-depth information on the implementation of E-Core will be given on the poster, e.g., internal structure and configuration parameters. Besides, information on our own NoC implementation and the application scenario is available and discussions are gladly welcome.

References

[1] U. Ogras, J. Hu, and R. Marculescu, “Key Research Problems in NoC Design: A Holistic Perspective,” in *Proc. of CODES+ISSS*, Jersey City, NJ, USA, September 2005.

[2] C. Cornelius and D. Timmermann, "Development and Operation of Networks-on-Chip," in *Proc. of the 14th VLSI-SoC*, Nice, France, October 2006.

[3] C. Cornelius, F. Sill, and D. Timmermann, "High-Level Simulations of On-Chip Networks," in *Proc. of the 9th DSD*, Cavtat, Croatia, August 2006.

[4] D. Wiklund, S. Sathe, and D. Liu, "Network on Chip Simulations for Benchmarking," in *Proc. of the 4th IEEE IWSOC*, Banff, Alberta, Canada, July 2004.

[5] S. Mahadevan et al., "A Network Traffic Generator Model for Fast Network-on-Chip Simulation," in *Proc. of the DATE'05*, Munich, Germany, March 2005.

[6] G. Palermo and C. Silvano, "PIRATE: A Framework for Power/Performance Exploration of Network-on-Chip Architectures," in *Proc. of the 14th PATMOS*, Santorini, Greece, September 2004.

[7] J. Xi and P. Zhong, "A System-level Network-on-Chip Simulation Framework Integrated with Low-level Analytical models," in *Proc. of the 24th ICCD*, San Jose, CA, USA, October 2006.

[8] N.Genko and D.Atienza and G. De Micheli et al., "A Complete Network-on-Chip Emulation Framework," in *Proc. of DATE'05*, Munich, Germany, March 2005.

[9] A. Jalabert, S. Murali, L. Benini, and G. D. Micheli, "xpipesCompiler: A Tool for Instantiating Application Specific Networks on Chip," in *Proc. of DATE'04*, Paris, France, February 2004.

[10] H. Widiger, S. Kubisch, D. Duchow, D. Timmermann, and T. Bahls, "An integrated Hardware Solution for MAT, MPLS-UNI, and TM in Access Networks," in *Proc. of the 31st IEEE LCN*, Tampa, FL, USA, November 2006.

[11] F. Gont, "ICMP attacks against TCP," October 2006, Internet Draft.

[12] F. Baker, "Requirements for IP Version 4 Routers," June 1995, RFC 1812.