Adv. Radio Sci., 6, 145–150, 2008
www.adv-radio-sci.net/6/145/2008/

Advances in Radio Science

Quantitative design space exploration of routing-switches for Network-on-Chip

M. C. Neuenhahn, H. Blume, and T. G. Noll

Chair of Electrical Engineering and Computer Systems, RWTH Aachen University, 52062 Aachen, Germany

Abstract. Future Systems-on-Chip (SoC) will consist of many embedded functional units like e.g. embedded processor cores, memories or FPGA like structures. These SoCs will have huge communication demands, which can not be fulfilled by bus-based communication systems. Possible solutions to this problem are so called Networks-on-Chip (NoC).

These NoCs basically consist of network-interfaces which integrate functional units into the NoC and routing-switches which connect the network-interfaces. Here, VLSI-based routing-switch implementations are presented. The characteristics of these NoCs like performance and costs (e.g. silicon area respectively logic elements, power dissipation) depend on a variety of parameters. As a routing-switch is a key component of a NoC, the costs and performance of routing-switches are compared for different parameter combinations. Evaluated parameters are for example data word length, architecture of the routing-switch (parallel vs. centralized implementation) and routing-algorithm.

The performance and costs of routing-switches were evaluated using an FPGA-based NoC-emulator. In addition different routing-switches were implemented using a 90 nm standard-cell library to determine the maximum clock frequency, power-dissipation and area of a VLSI-implementation. The power consumption was determined by simulating the extracted layout of the routing-switches. Finally, these results are benchmarked to other routing-switch implementations like Aetheral and xpipes.

1 Introduction

The complexity of VLSI-chips is increasing more and more due to recent improvements in VLSI-technology (ITRS, 2005). As the requirements given by applications concerning e.g. computing power and/or power-dissipation are increasing, VLSI-chips more and more turn to so called Systems-on-Chip. These SoCs consist of functional units like for example processor-cores, memories, DSPs or embedded FPGAs. The number of functional units in these SoCs is constantly increasing (Blume et al., 2005) and is supposed to reach the mark of hundred in the near future (Borkar et al., 2005).

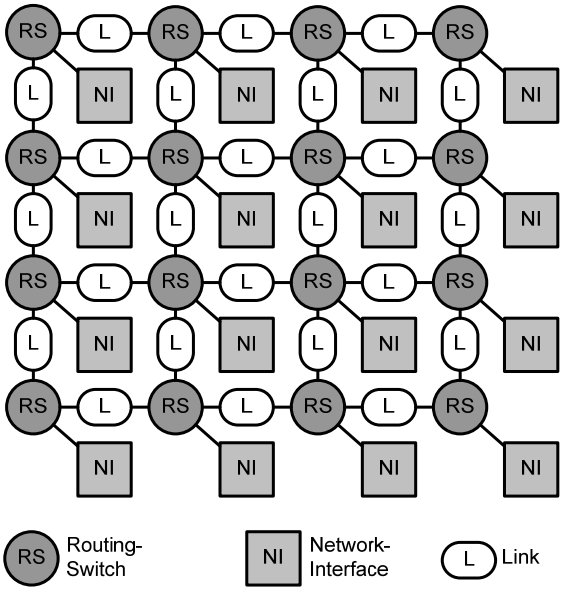

Fig. 1. NoC with mesh topology.

The communication demands of applications mapped onto these complex SoCs will be very high. For example, high throughput shall be provided at minimal latency, with guaranteed quality of service (QoS) and as much flexibility as possible. Furthermore, the communication should cause minimal costs, in terms of area and power consumption. As the number of functional units in SoCs will increase in the future the communication-structure should although be scalable. Traditional communication structures like on-Chip busses or point-to-point connections cannot fulfill these demands. Often quoted solutions to this problem are Network-on-Chip architectures (Benini and De Micheli, 2002).

Correspondence to: M. C. Neuenhahn
(neuenh@eecs.rwth-aachen.de)

Published by Copernicus Publications on behalf of the URSI Landesausschuss in der Bundesrepublik Deutschland e.V.

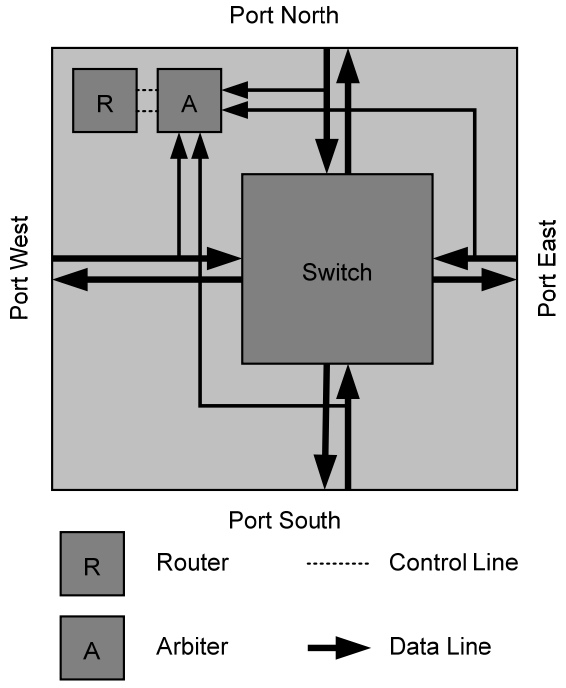

Fig. 2. Centralized architecture of a routing-switch with 4 ports.

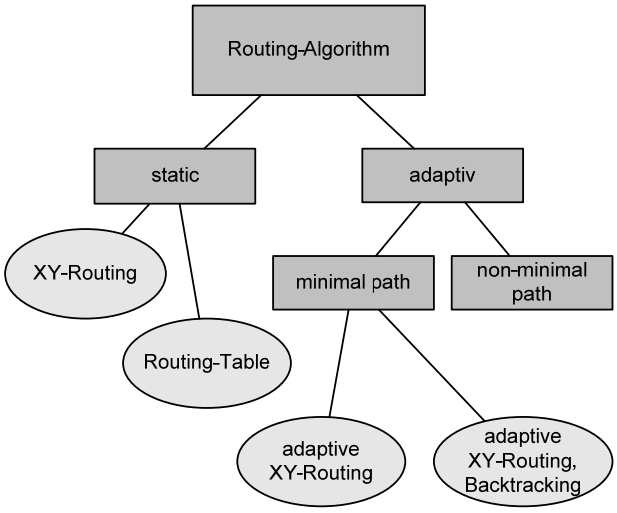

Fig. 3. Overview of routing-algorithms for Network-on-Chip.

The performance and costs caused by these NoCs strongly depend on NoC-specific parameters like topology (Neuenhahn et al., 2006), data word length, routing-algorithm and many more (Bjerregaard and Mahadevan, 2006). These NoC-parameters span a huge design-space for NoCs. As each application or application-class has different communication requirements the appropriate NoC-parameter combination that fulfills these requirements at minimal costs has to be found (Ahonen et al., 2005; Benini, 2006).

The implementation effort of NoCs with an altered parameter set, e.g. modified topology or data word length, should be low. Therefore, a modular and generic NoC-architecture like e.g. xpipes (Bertozzi and Benini, 2004) or the NoC-architecture presented in Sect. 2 is needed. Such generic architectures allow a modification of single NoC-parameters at minimum design effort.

A key-component of NoCs concerning area, energy dissipation and performance is the routing-switch. Hence, a detailed analysis of implementation costs and influence on performance of routing-switches featuring different NoC-specific parameters is needed.

The NoC-parameter routing-algorithm has great impact on the performance of NoCs. Many different algorithms for computer networks and NoCs have been proposed and evaluated (Duato et al., 2003; Moraes et al., 2004). But these evaluations did not regard the impact on implementation costs and focus only on performance aspects.

The architecture of routing-switches and the functionality of its building blocks are presented in Sect. 2. An evaluation of the performance of these routing-switches and implementation results concerning area and power are presented in Sect. 3. The NoC-parameters data word length, routing-algorithm and number of ports have been varied in cause of the experiments. Finally, an implemented routing-switch is compared to routing-switches recently published.

In Sect. 4 the design space of routing-switches is discussed, before in Sect. 5 a conclusion is given.

2 Modular and generic Network-on-Chip

The modular and generic NoC framework which was introduced in Neuenhahn et al. (2006) features three basic building blocks: network-interface, routing-switch and link. The network-interface is the gateway between NoC and a functional unit, the routing-switch routes data through the NoC and the link represents the connections between these building blocks. The arrangement of these blocks is defined by the NoC-parameter topology. The options of possible topologies range from classic topologies like mesh (see Fig. 1), torus or star topologies to application specific hierarchical topologies.

Every NoC component, network-interface, routing-switch and link, consists of different modules. Each of these modules can be implemented in different ways to realize a defined functionality. For example the module which is used for applying error-correcting or error-detecting codes in network-interfaces or routing-switches can be implemented using no coding, parity code or Huffman-codes (Neuenhahn et al., 2007).

This modular and generic approach enables the implementation of NoC with a variety of NoC-specific parameters at minimal design effort. In Neuenhahn et al. (2007) a NoC design-flow based on this framework was presented.

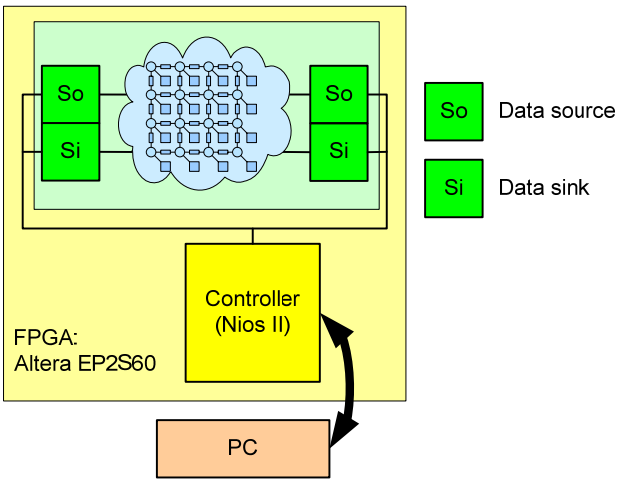

Fig. 4. FPGA-based emulation environment.

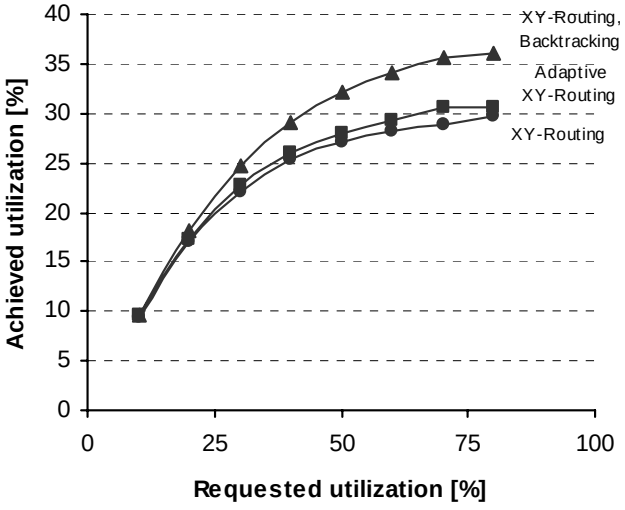

Fig. 5. Performance of different routing-algorithms for a NoC with Mesh topology and 16 functional units.

A routing-switch for NoCs consists of a switch, which connects the input ports with corresponding output ports. This switch is controlled by an arbiter, which grants or refuses the access to an output port. The routing-decision, to which output port data should be routed, is made in the router. This decision is based on the routing-algorithm, the position of the routing-switch in the network and the target address of the data. In case of adaptive routing-algorithms, the state of the routing-switch, e.g. used ports, is regarded as well. Here, circuit switching is used as switching technique and therefore no buffers are needed inside a routing-switch.

Different routing-switch architectures have been analyzed. The centralized architecture presented in Fig. 2 features the smallest area at negligible performance decrease compared to a distributed architecture. Depending on the timing-requirements, additional registers at the input ports can be applied to perform pipelining.

Routing-algorithms for NoCs can be distinguished in static and adaptive routing-algorithms, as depicted in Fig. 3. Static algorithms route the data always on the same path from the source to the destination. A widely used class of static routing-algorithms is dimension-order routing. In a NoC with mesh topology, as depicted in Fig. 1, such an algorithm is XY-routing. The data is at first routed in X-direction and afterwards in Y-direction over the NoC. The state of a routing-switch, e.g. if an output port is already used, is not regarded. In this case data transmission is aborted and reinitiated by the network-interface at a later point of time. The routing-algorithm can be implemented as a state-machine or as a routing-table.

Adaptive routing-algorithms can be distinguished in algorithms supporting only minimal paths and algorithms supporting non-minimal paths. The latter algorithms are not regarded in the following evaluation.

The adaptive XY-routing-algorithm routes data at first in X-direction. If an output port determined by the XY-routing is already used, an alternative output port is chosen, which leads the data directly towards the target. If no feasible output port can be determined, the data transmission is cancelled and reinitiated by the sending network-interface at a later point of time.

Using adaptive XY-routing with backtracking the data transmission is not cancelled when all feasible output ports of a routing-switch are blocked. Instead of that, another routing attempt at the previous routing-switch, if present, is performed.

3 Performance and implementation results

To evaluate the influence of the routing-algorithm on the performance of NoC an FPGA-based emulator (Neuenhahn et al., 2006) was used. This FPGA-based NoC-Emulator emulates a NoC on an FPGA. Even on modern FPGAs it is difficult to handle the complexity which is required for emulating a complex SoC. To solve this problem, the functional units of a SoC e.g. processor kernels or memories are replaced with abstract data sources and data sinks. With this abstracted functional units the NoC can be emulated.

The FPGA-based generic emulation environment depicted in Fig. 4 consists of an FPGA and a PC. The PC is used to synthesize the FPGA-configuration, initiate experiments and evaluate the results of the experiment. The FPGA contains the NoC to be emulated (see Sect. 2), traffic-sources and -sinks which emulate functional units of a SoC and a soft-core processor (Altera Corporation, 2007) which is used to control the data-sources and evaluate the raw results generated by the data-sinks.

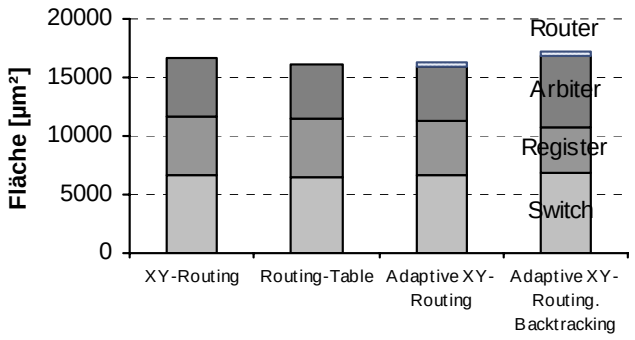

Fig. 6. Area distribution of routing-switches with 5 ports, 32 bit data word length, registers at the input port and varied routing-algorithms.

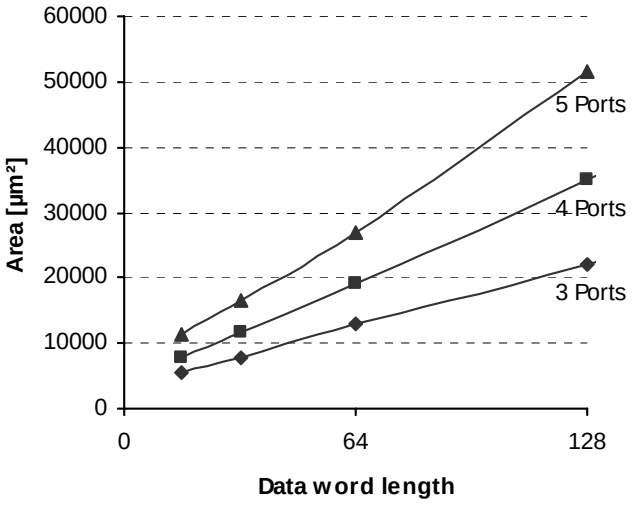

Fig. 7. Total area of a routing-switch with XY-routing and input port registers for varied data word length and different numbers of ports.

As depicted in Fig. 5, for a requested utilization smaller than 20% no influence on the performance of the NoC depending on the routing-algorithm occurs. Applying a higher utilization, adaptive algorithms result in a higher performance than the static routing-algorithms. The adaptive XY-routing with backtracking has obviously the highest performance. For a requested utilization the achieved utilization is about 15% higher than the achieved utilization obtained by the other algorithms.

Further experiments using other topologies and different NoC-sizes show that the differences in performance depending on these routing-algorithms increase for bigger and decrease for smaller NoCs.

The area needed for a VLSI-implementation of a routing-switch was obtained using the following design flow. At first, the VHDL-code describing the routing-switch was synthesized using Synopsys Design Compiler (Synopsys, 2007). The resulting netlist was used in the place-and-route tool Encounter (Mentor Graphics, 2006). For verification and power evaluation the resulting layout has been extracted into a SPICE-netlist (Cadence, 2006) which has been simulated using NanoSim (Synopsys, 2006). A 90 nm technology from TSMC and a standard-cell library from Artisan Components (Artisan Components Inc., 2005) has been used. The links, connected to each routing-switch are assumed to have a length of 1 mm. The resulting capacity on each output port is calculated to 0.148 pF (5 $\times$ minimum distance to neighbors, toggling against each other) and the operating frequency is 1 GHz. All results are computed using slow operating conditions.

As depicted in Fig. 6 the differences in area dissipation for routing-switches with 5 ports, 32 bit data word length and additional registers at the input ports featuring different routing-algorithms is less than 5%. The total area needed for control, router and arbiter, is about 20% of the total routing-switch area.

The correlation of routing-switch area, data word length and number of ports is presented in Fig. 7. There is a linear dependency of the area and the data word length. The gradient of the linear area function is determined by the number of ports. The area consumed by registers and the switch block increases with higher data word length whereas the area needed for router and arbiter remains constant.

The power consumption depends on the state of the routing-switch. As in this work circuit switching is performed, the routing-switch/output ports can be in the following states: “idle”, “determine connection target”, “wait for connection to establish”, “active connection” and “destroy connection”.

In the state “idle” no action is performed, while in the state “determine connection target” the arbiter and router determine the target port of the connection. As this state lasts only three cycles, the contribution to the overall energy-consumption is negligible. The state “wait for connection to establish” has approximately the same power consumption than the “idle” state. The highest power consumption and the dominant part in energy needed for a data transmission is used in the state “active connection”. As the state “destroy connection” lasts only two clock-cycles, its share to the overall energy of a data transmission is negligible, too.

The power consumption of a single routing-switch is shown in Fig. 8. It is in the state “active connection” with one active connection. It depends nearly linear on the data word length, whereas the gradient is defined by the number of ports. A routing-switch can feature several connections simultaneously. Hence, these states can be superposed by other active connections.

To compare this routing-switch to implementations published, the area and maximum clock-frequency have been scaled to a 90 nm technology. In addition to that, the area has been normalized to one bit data word length, as

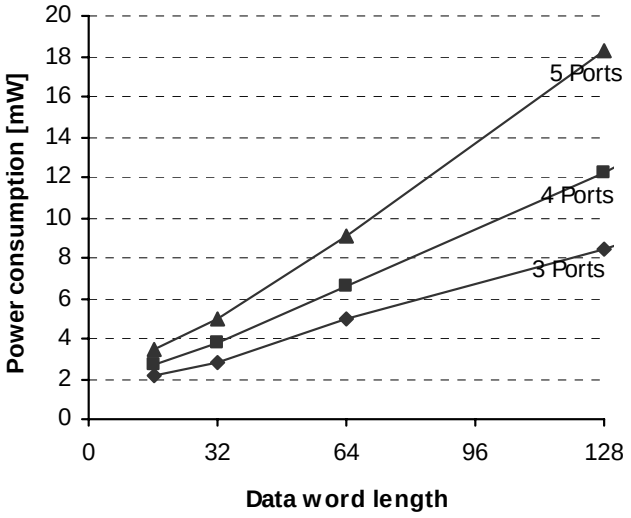

Fig. 8. Total power consumption of a routing-switch with XY-routing and input port registers for varied data word length and different numbers of ports with one active connection and a toggle-rate of approximately 25%.

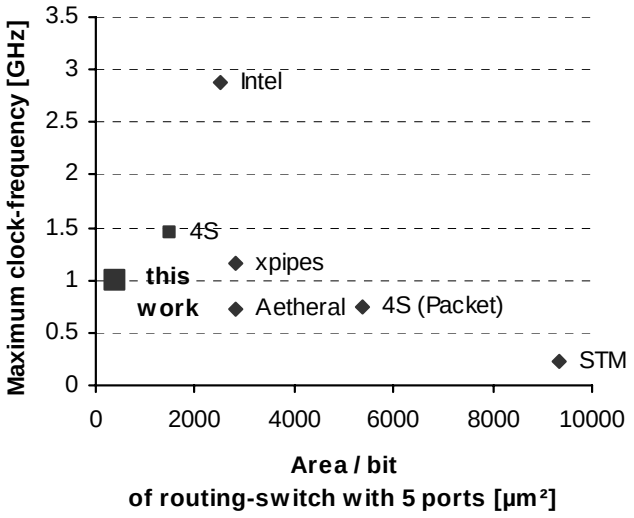

Fig. 9. Comparison of routing-switch implementations concerning area and maximal clock frequency. The area and maximum clock-frequency was scaled to a 90 nm technology.

the implementations presented feature data word lengths from 16 to 32 bit. The values are taken from Rijpkema et al. (2003)(Aetheral), Vangal et al. (2007) (Intel), Lattard et al. (2007)(STM), Stergiou et al. (2005) xpipes and Wolkotte et al. (2005)(4S and 4s(Packet)). These values are enlisted in Fig. 9.

The comparison in Fig. 9 shows that the implementation described here has the smallest area consumption. These results are based on the fact that in most implementations, besides the implementation named 4S, packet switching is performed, while in this work circuit switching is performed. Hence, no buffers are needed in this routing-switches resulting in smaller area.

The implementations 4S, xpipes and Intel outperform the implementation presented in this work concerning the maximum clock-frequency, but this implementation was optimized for the clock-frequency of 1 GHz. A further enhancement can easily be achieved by modifying the synthesis parameter at the cost of a slightly increased area.

4 Design space of routing-switches for NoC

In Fig. 10 different routing-switch implementations have been evaluated concerning the area used and the performance a NoC featuring these routing-switches achieves. The NoC has a mesh topology with 16 functional units attached. It features circuit switching and no error protection at data word level.

The data word length was varied from 16 to 256 bit; the routing-algorithms presented were used and input ports with and without registers were applied.

To achieve performance figures in terms of average latency, several experiments have been carried out using the FPGA-based emulator. Benchmarks based on random data traffic with a requested data rate of 204.8 GBit/s were applied to the various NoC instantiations. The NoC implemented with standard cells should operate at a clock-frequency of 1 GHz.

The dots in Fig. 10 representing routing-switch implementations result in a pareto-front. For each of these implementations laying on the pareto-front no other implementation exists featuring a higher performance (lower average latency) at lower or equal implementation costs, expressed in area. For a given combination of requested data rate and allowed average latency the NoC-parameter set fulfilling these requirements at minimal routing-switch area can be identified using this diagram.

5 Conclusion

In this work routing-switch implementations for different NoC-specific parameter combinations have been presented. It has been shown that the routing-algorithm has a significant impact on the performance and only a small impact on the resulting costs, especially for NoCs with a high data word length. Hence, the adaptive XY-routing with backtracking is the favorable routing-algorithm in most cases.

Furthermore, the dependency of area and power of routing-switches on the parameters data word length and port number has been presented. Compared to routing-switch implementations published, these implementations feature a much smaller area. The presented design space exploration is essential for determining an efficient NoC-specific parameter combination for given communication demands.

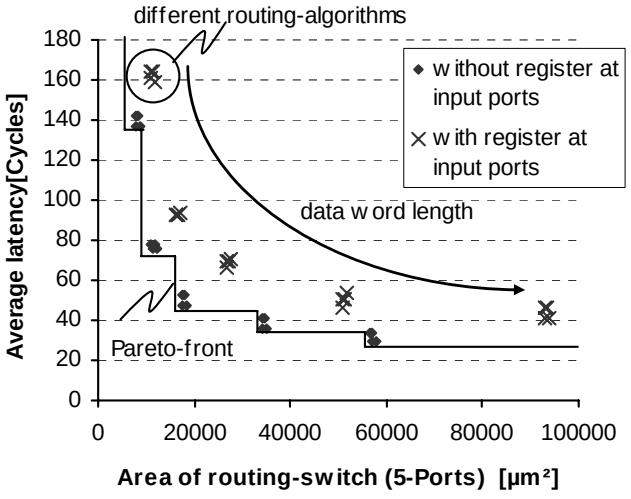

Fig. 10. Pareto-plot of routing-switch implementations.

References

Ahonen, T., Virtanen, S., Kylliäainen, J., Truscan, D., Kasanko, T., Sigäuenza-Tortosa, D., Ristimäaki, T., Paakkulainen, J., Nurmi, T., Saastamoinen, I., Isäannäainen, H., Lilius, J., Nurmi, J., and Isoaho, J.: A Brunch from the Coffee Table-Case Study in NoC Platform Design, Interconnect-Centric Design for Advanced SoC and NoC, 425–453, 2005.

Altera Corporation: Nios II Processor Reference Handbook, 2007.

Artisan Components, Inc.: TSMC 90nm CLN90G Process SAGE-X v3.0 Standard Cell Library Databook, 2005.

Benini, L.: Application Specific NoC Design, Design, Automation and Test in Europe, DATE '06, 2006.

Benini, L. and De Micheli, G.: Networks on chips: a new SoC paradigm, Computer, 35(1), 70–78, 2002.

Bertozzi, D. and Benini, L.: Xpipes: a network-on-chip architecture for gigascale systems-on-chip, Circuits and Systems Magazine, IEEE 4(2), 18–31, 2004.

Bjerregaard, T. and Mahadevan, S.: A survey of research and practices of Network-on-chip, ACM Comput. Surv., 38(1), 1–51, 2006.

Blume, H., Feldkaemper, H. T., and Noll, T. G.: Model-Based Exploration of the Design Space for Heterogeneous Systems on Chip, The Journal of VLSI Signal Processing, 40(1), 19–34, 2005.

Borkar, S. Y., Mulder, H., Dubey, P., Pawlowski, S. S., Kahn, K. C., Rattner, J. R., and Kuck, D. J.: Intel: Platform 2015: Intel® Processor and Platform Evolution for the Next Decade, 2005.

Cadence: Design Framework, 2006.

Duato, J., Yalamanchili, S., et al.: Interconnection Networks – An Engineering Approach, San Francisco, USA, Morgan Kaufmann Publishers, 2003.

ITRS: International Technology Roadmap for Semiconductors, 2005 Edition, Executive Summary, 2005.

Lattard, D., Beigne, E., Bernard, C., Bour, C., Clermidy, F., Durand, Y., Durupt, J., Varreau, D., Vivet, P., Penard, P., Bouttier, A., and Berens, F.: A Telecom Baseband Circuit based on an Asynchronous Network-on-Chip, Solid-State Circuits Conference, 2007, ISSCC 2007, Digest of Technical Papers, IEEE International, 2007.

Mentor Graphics: Encounter, 2006.

Moraes, F., Calazans, N., Mello, A., Moller, L., and Ost, L.: HERMES: an infrastructure for low area overhead packet-switching networks on chip, Integration, the VLSI Journal, 38(1), 69–93, 2004.

Neuenhahn, M. C., Blume, H., and Noll, T. G.: Quantitative analysis of network topologies for NoC-architectures on an FPGA-based emulator, Presentation, Kleinheubacher Tagung, 2006.

Neuenhahn, M. C., Lemmer, D., Blume, H., and Noll, T. G.: Quantitative Cost Modeling of Error Protection for Network-on-Chip, ProRISC Workshop, Veldhoven, 2007.

Rijpkema, E., Goossens, K. G. W., Radulescu, A., Dielissen, J., Meerbergen, J. v., Wielage, P., and Waterlander, E.: Trade Offs in the Design of a Router with Both Guaranteed and Best-Effort Services for Networks on Chip, Proceedings of the conference on Design, Automation and Test in Europe – Volume 1, 2003.

Stergiou, S., Angiolini, F., Carta, S., Raffo, L., Bertozzi, D., and De Micheli, G.: Xpipes Lite: a synthesis oriented design library for networks on chips, Design, Automation and Test in Europe, Proceedings, 2005.

Synopsys: NanoSim, 2006.

Synopsys: Design Compiler, 2007.

Vangal, S., Howard, J., Ruhl, G., Dighe, S., Wilson, H., Tschanz, J., Finan, D., Iyer, P., Singh, A., Jacob, T., Jain, S., Venkataraman, S., Hoskote, Y., and Borkar, N.: An 80-Tile 1.28TFLOPS Network-on-Chip in 65nm CMOS, Solid-State Circuits Conference, 2007, ISSCC 2007, Digest of Technical Papers, IEEE International, 2007.

Wolkotte, P. T., Smit, G. J. M., Rauwerda, G. K., and Smit, L. T.: An Energy-Efficient Reconfigurable Circuit-Switched Network-on-Chip, Parallel and Distributed Processing Symposium, Proceedings 19th IEEE International, 2005.