

Realtime multiprocessor for mobile ad hoc networks

T. Jungeblut, M. Grünewald, M. Pormann, and U. Rückert

System and Circuit Technology, Heinz Nixdorf Institute, University of Paderborn, Germany

Abstract. This paper introduces a real-time Multiprocessor System-On-Chip (MPSoC) for low power wireless applications. The multiprocessor is based on eight 32bit RISC processors that are connected via an Network-On-Chip (NoC). The NoC follows a novel approach with guaranteed bandwidth to the application that meets hard realtime requirements. At a clock frequency of 100 MHz the total power consumption of the MPSoC that has been fabricated in 180 nm UMC standard cell technology is 772 mW.

1 Introduction

Portable electronic devices like PDAs, mobile phones and notebooks are increasingly equipped with wireless communication technologies, providing higher degrees of mobility and ease of use. Mobile ad hoc networks (MANETs) are a special type of wireless networks that do not require any infrastructure and whose topology can change spontaneously by the movement of participating nodes. To evaluate the performance and energy efficiency of new routing algorithms, especially including directional communication (Grünewald et al., 2005b) and transmission power control (Xu et al., 2005), we use the network simulator SAHNE (Volbert, 2002), see Fig. 1. This environment emulates the packet processing of each participating node. Simulations have shown that communicating in eight directions can increase the total throughput in a mobile ad hoc network by the factor of 2.5.

Efficient system components for medium access and routing are required for processing of the transmitted data packets. In our work, we evaluate Multiprocessor System-on-Chips (MPSoC) instead of single processors with higher clock frequency to achieve the required performance and resource efficiency. The MPSoC comes with an application specific “Network-on-Chip (NoC)” that offers a high bandwidth and guaranteed latency (Grünewald et al., 2005a). The protocol functions of the application are automatically

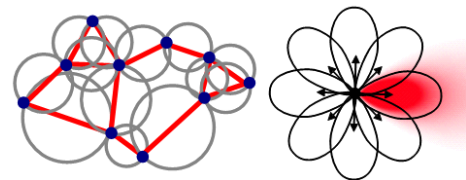


Fig. 1. Transmission power control and directional communication.

mapped to the processing elements via our software tool. The first MPSoC, which has been developed in our group, consists of eight processing elements (PEs), connected via the NoC. The communication with the peripherals is done via physical ports that are connected to air-to-air interfaces for the directional communication. Running a CSMA/CA-protocol, which we enhanced by the functionality of directional communication, the SoC can guarantee an external port data rate of 2.6 MBit/s and a total throughput of 21.6 Mbit/s on a system with eight physical ports, 16 processing elements and 16 switch boxes.

2 Architecture of the multiprocessor

With a high number of processing elements no common bus-based structures can be used. We use a homogeneous System-On-Chip (SoC) with a flexible NoC that is described in Sect. 2.1 in detail. The NoC connects the processing elements (PE) and the switch boxes (SB) to form the SoC. The physical ports (PP) are used for off-chip-communication. As our SoC is described in VHDL, generic and scalable, different architectures can be implemented. The number of the switch-box can be adopted to the needed number of PEs or NoC-links and through a generic uplink- and downlink interface different softcore processing elements can be evaluated. Additionally the interface of the physical ports can be adjusted to the requirements of the surrounding system. One possible architecture of the SoC is shown in Fig. 2.



Correspondence to: T. Jungeblut
(tj@hni.upb.de)

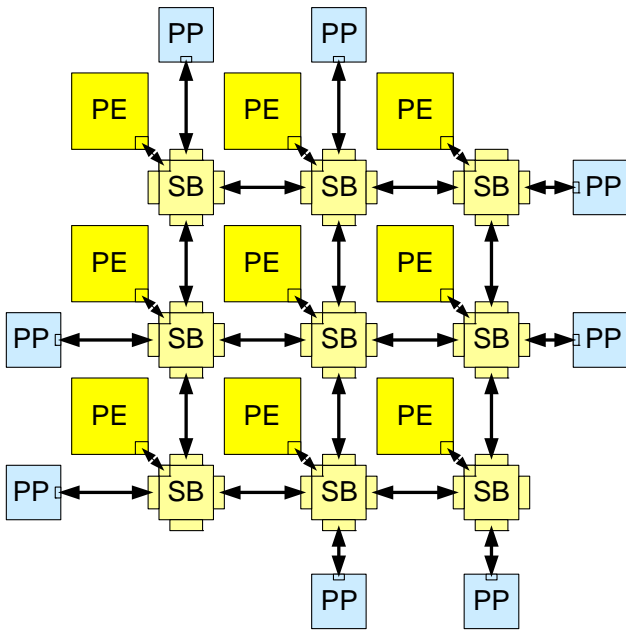


Fig. 2. Possible Architecture variant of the SoC.

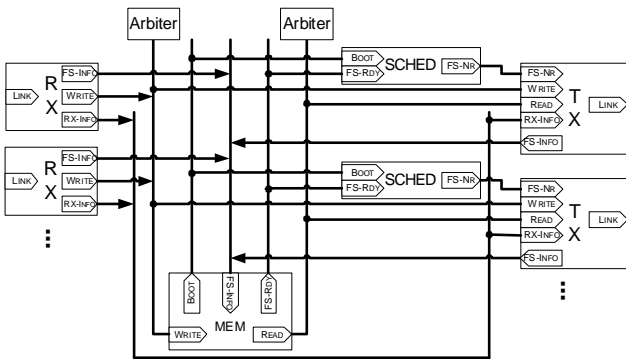


Fig. 3. Architecture of the switch box.

2.1 The Network-on-Chip

A specific feature of the multiprocessor is the predictability of its performance. Packet based communication is used instead of simple time multiplexing, enabling a high bandwidth utilization. In Grünewald et al. (2005a) methods have been proposed to assign the protocol functions to processors and to estimate the resource consumption of the final mapping. During the mapping either delay or energy consumption per packet can be minimized. An essential requirement for this methodology is that the upper bound for the latency of a packet can be calculated. As usual in NoCs, the packets are divided into data units of fixed size, called flits.

Figure 3 shows the switch box of the NoC. The SB consists of receivers (RX), transmitters (TX), schedulers (SCHED),

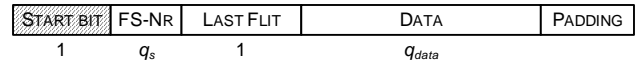


Fig. 4. Segmented flit for off-chip communication.

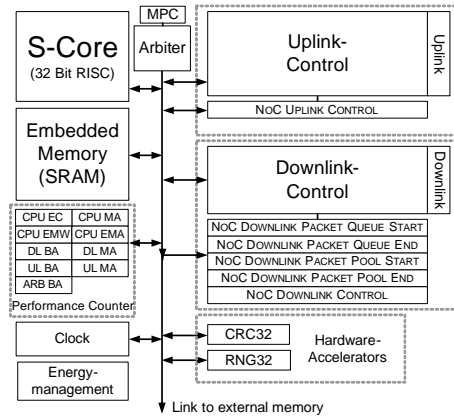


Fig. 5. Block diagram of the processing element.

and flit-memory. Receiver and transmitter write/read flits to/from a shared memory which is called shared memory switching. For latency reduction, the TX-, SCHED-, and RX-units are working in parallel while receiving a flit. With the given number of links of the SB (N_{SB}), the total number of execution cycles $EC_{SB,rx}$ for one reception cycle is:

$$1 \leq EC_{SB,rx} \leq N_{SB} \tag{1}$$

The transmit cycle starts as soon as the flow control detects an incoming flit, which has to be forwarded via the corresponding output-link. In worst case, i.e., if all transmitters detect a transmission request, the memory responds after N_{SB} cycles. With additional three clock cycles for receiving the flit from the scheduler, storing it in an internal temporary register and forwarding it to the output register, the total number of execution cycles $EC_{SB,tx}$ is:

$$3 \leq EC_{SB,tx} \leq N_{SB} + 3 \tag{2}$$

The first flits, the SBs receive during system start up, are boot-flits, which are used to initialize the routing tables. For communication with the surrounding system, the physical port of our MPSoC segments the flits to reduce the number of required I/O-pins. Figure 4 shows the structure of the segmented flits. The length l of a segmented flit is given by $l=q_s+q_{data}+2$, where q_s is the index of the associated flow segment, which represents a virtual connection, and q_{data} is the number of data bits.

2.2 The Processing Element

Figure 5 shows a block diagram of the processing element that is used in our MPSoC. Central component of the

proposed architecture is the S-Core-processor (Langen et al., 2002), which has been developed in our group. S-Core is a 32 Bit-RISC-processor with a three-stage-pipeline, instruction-set compatible to the Motorola M-Core. 32 kB local static memory per PE can be used for instructions and data. Additionally, external memory can be accessed to execute memory intensive tasks. Furthermore, the PEs are equipped with CRC hardware accelerators, a timer, and a random numbers generator. Via uplink interfaces and downlink interfaces the PEs are connected to the Network-on-Chip. The number of execution cycles EC_{DL} and EC_{UL} for receiving and transmitting a flit is given by:

$$q_{data}/32 + 4 \leq EC_{DL} \leq q_{data}/32 + 6 \quad (3)$$

and

$$q_{data}/32 + 6 \leq EC_{UL} \leq q_{data}/32 + 8 \quad (4)$$

To determine the resource efficiency of the software and of the hardware implementation, our VHDL-based characterisation environment PERFMON is used (Grünwald et al., 2005a). PERFMON provides an infrastructure for simulation and evaluation of the whole system, including main memory, debugging units, and performance counters. Each software processing element can be used to substitute the currently used S-Core and to analyze its performance in the proposed multiprocessor environment. Because all parts of PERFMON are synthesizable and generic, the whole system can be mapped to a hardware technology like an ASIC or an FPGA for rapid prototyping. Target specific components are replaced automatically.

2.3 Prototyping environment

Initial implementations of the system have been mapped to FPGA architectures and have been tested in our rapid prototyping environment RAPTOR2000 (Kalte et al., 2002) (Fig. 7). This allows for fast simulation and verification in early design stages. As a proof-of-concept, the multiprocessor is integrated in the SAHNE simulation environment (Volbert, 2002), which is used to simulate the nodes of a mobile ad-hoc-network. Packet processing of one node is not simulated, but executed on the MPSoC architecture. The hardware is connected to the simulator using the hardware abstraction layer (HAL) of the packet processing library, which has been presented in Grünwald et al. (2005a). The HAL also ensures the synchronization of the hardware and the simulator. By this hardware/software co-simulation of large mobile ad-hoc-networks are possible.

2.4 The multiprocessor ASIC

After successful testing, the FPGA prototype is replaced by an fabricated ASIC. Because of the modular approach of the

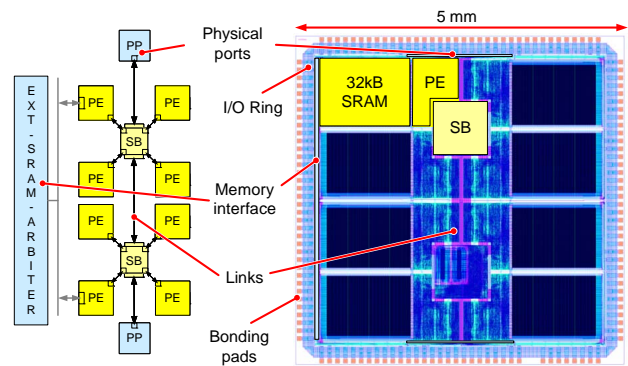


Fig. 6. Chip photo of the MPSoC.

RAPTOR2000 rapid prototyping system, the test environment can be reused as described in section 2.4.1. The hierarchical design of the ASIC shortens development time, because parts as the processing elements and the switch boxes have to be designed only once and then can be multiply instantiated. This is also an advantage of our SoCs as wire-length can be calculated better in advance and more aggressive signaling strategies can be used. In the fabricated multiprocessor (see Fig. 6) four processing engines are connected to one switch box. Two of these processor clusters form the eight-core MPSoC. This architecture results from a design space exploration and simulation of different architectures as described in Sect. 1 and achieves a high resource efficiency which is important for low power applications as mobile ad hoc networks. The proposed system has been manufactured in 180 nm UMC standard-cell-technology and occupies an area of 25 mm² using six metal layers. It embeds 2.1 MBit memory and consists of 1.6 million transistors. At a clock frequency of 100 MHz, the average power consumption is 772 mW. At this speed, a communication bandwidth of up to 2.1 Gbps is achieved for each link of the NoC. 4.2 Gbps throughput per switch box are achieved in total with all six links active which is a disadvantage of memory shared switching. The off-chip communication bandwidth via two physical ports is 500 MBit/s. A daughterboard for RAPTOR2000 has been developed, comprising the MPSoC, 4 MB external memory, and a Spartan XC3S1500 FPGA, integrating an interface to the RAPTOR2000 motherboard (see Fig. 7). The user can easily interact with the MPSoC, using the PCI bus interface of RAPTOR2000.

2.4.1 Testing the SoC

This prototyping environment is also intended to test the functionality of new chip charges. On the host-system a monitor program controls the initialization of the processing elements and the incoming and outgoing traffic via the physical ports. Once the switch boxes are initialized, the memory images of each processing element is sent to the on-chip memory via

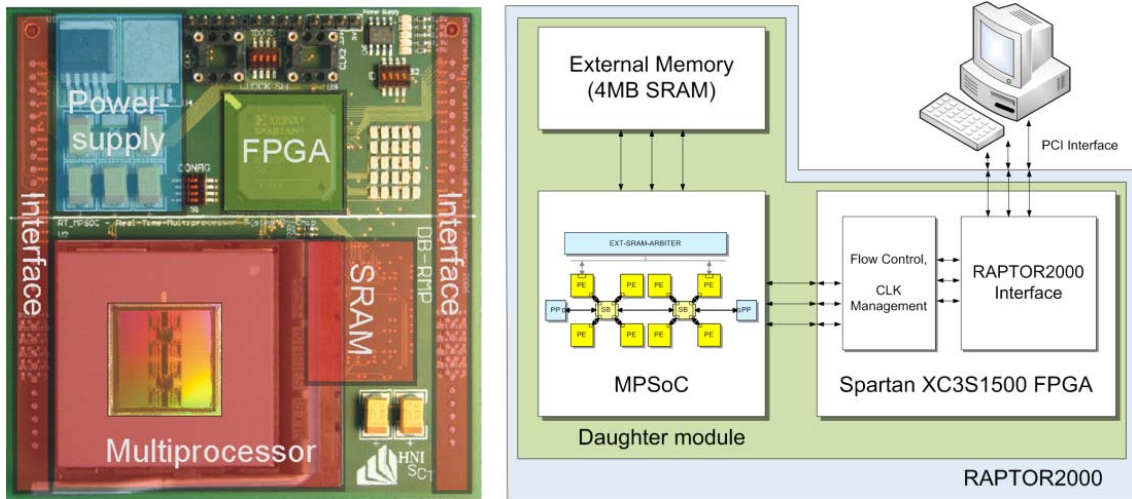


Fig. 7. Test-Environment for the MPSoC.

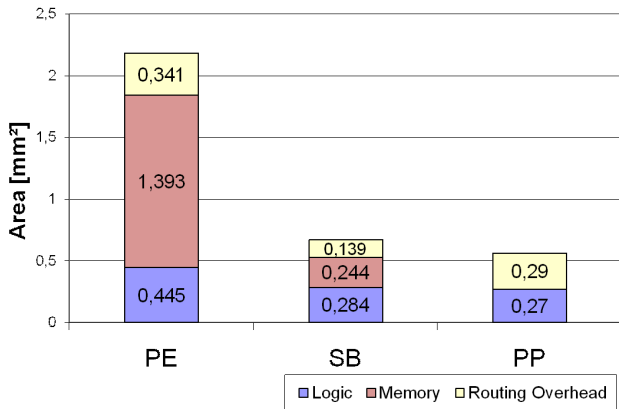


Fig. 8. Area requirements of the core components of the MPSoC.

the NoC. We developed different test cases to test the functionality of each processing element, the on-chip memories, the switch-boxes, the interface to external memory, the NoC-links and the physical ports. To verify the correct behavior of the system, these test results can be automatically compared with those of the simulation and FPGA emulation. To determine the bottlenecks of the system or the distribution in the fabrication of the ASICs, we need to operate the components of the SoC at different clock frequencies. The interface on the additional FPGA enables the variation of the clock frequency during runtime. In this way we can operate the NoC at a specific speed while transmitting packets via the NoC and afterwards switching to a different frequency to determine the maximum performance of the processing engines.

Table 1. Power dissipation of the MPSoC [mW].

	Highest Load		Idle	
	8xPE	2xSB, 2xPP	8xPE	2xSB, 2xPP
PE	74,254	35,864	594,032	286,912
SB	86,288	24,417	172,576	48,834
PP	2,6	1,451	5,2	2,902
Total			771,808	338,648

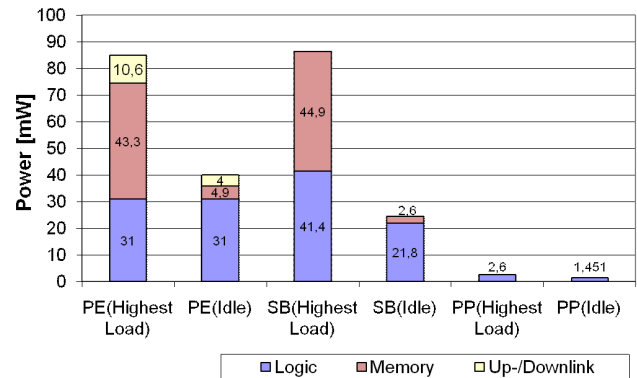


Fig. 9. Power dissipation of the core components of the MPSoC.

2.5 Ressource consumption

Figure 8 shows the area consumption of the core components of the MPSoC. The largest part is the processing element (2.18 mm²), basically because of the large on-chip static memory (1.38 mm²=64%). The switch box and the physical port uses less than one third of the area of one PE.

As there are only two of each in contrast to eight PEs, their impact on the total area is insignificant in the realized architecture variant. Figure 9 shows the power dissipation for idle state and highest load. Because currently no power management is used in the processing elements, there is only an reduction of the power of the on-chip memory. With an intelligent power management, at different levels of hierarchy (Clock gating, gating of unused functional blocks in the PEs, gating of entire PEs) energy could be saved. Table 1 shows the simulated power consumption of the entire SoC determined from Synopsys Power Compiler. The switching activities caused by the CSMA/CA protocol were annotated to get more accurate results. The measured power consumption of 470 mW is below the simulated value of 772 mW probably because of the worst case assumption of the Synopsys tools. As before, the highest impact on the total power consumption is caused by the eight processing elements.

3 Conclusions

In this work we have presented a generic, scalable architecture for Multiprocessor SoCs (MPSoC) intended for low power wireless applications as mobile ad hoc networks. Via generic uplink- and downlink interfaces, standard IP-cores can be used as processing elements. The NoC follows a novel approach to guarantee minimum bandwidth to the application to meet hard realtime requirements. An FPGA-based-prototype is used for fast hardware-software-co-simulation of mobile ad hoc networks. An eight-core MPSoC-chip prototype has been fabricated. The proposed ASIC has been manufactured in 180 nm UMC standard-cell-technology and occupies an area of 25 mm² at a power consumption of 772 mW.

References

- Grünewald, M., Niemann, J.-C., Pörrmann, M., and Rückert, U.: A framework for design space exploration of resource efficient network processing on multiprocessor SoCs, Morgan Kaufmann Publishers, 3(12), 245–277, 2005a.
- Grünewald, M., Xu, F., and Rückert, U.: Increasing the Resource – Efficiency of the CSMA/CA Protocol in Directional Ad Hoc Networks, in: Proceedings of the 4th International Conference on AD-HOC Networks & Wireless, Cancun, Mexico, 71–84, 2005b.
- Kalte, H., Pörrmann, M., and Rückert, U.: A Prototyping Platform for Dynamically Reconfigurable System on Chip Designs, in: Proc. of the IEEE Workshop Heterogeneous reconfigurable Systems on Chip (SoC), Hamburg, Germany, 2002.
- Langen, D., Niemann, J.-C., Pörrmann, M., Kalte, H., and Rückert, U.: Implementation of a RISC Processor Core for SoC Designs – FPGA Prototype vs. ASIC Implementation, in: Proceedings of the IEEE-Workshop: Heterogeneous reconfigurable Systems on Chip (SoC), Hamburg, Germany, 2002.
- Volbert, K.: A Simulation Environment for Ad Hoc Networks Using Sector Subdivision, in: Proceedings of the 10th Euromicro Workshop on Parallel, Distributed and Network-based Processing, Canary Islands, Spain, 419–426, 2002.
- Xu, F., Grünewald, M., and Rückert, U.: A Low Complexity Directional Scheme for Mobile Ad Hoc Networks, in: Proceedings of the 16th IEEE International Symposium on Personal Indoor and Mobile Radio Communications, 1349–1353, 2005.