Real-Time Multiprocessor SoC for Mobile Ad Hoc Networks

Thorsten Jungeblut, Matthias Grünewald, Mario Porrmann, Ulrich Rückert

{tj, matthias.gruenewald, porrmann, rueckert}@hni.upb.de

University of Paderborn – System and Circuit Technology – Germany

http://wwwhni.upb.de/en/sct

Abstract

This paper introduces our Real-time Multiprocessor SoC intended for low power wireless applications as mobile ad hoc networks. The multiprocessor is based on eight of our 32bit S-CORE RISC processors.

1. Introduction

Portable electronic devices like PDAs, mobile phones and notebooks are increasingly equipped with wireless communication technologies, enabling 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 and transmission power control, we use the network simulator SAHNE [2]. 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) to achieve the required resource efficiency. The first MPSoC, which has been developed in our group, consists of eight processing engines (PEs), connected via an application-specific Network-on-Chip (NoC), as shown in Figure 1.

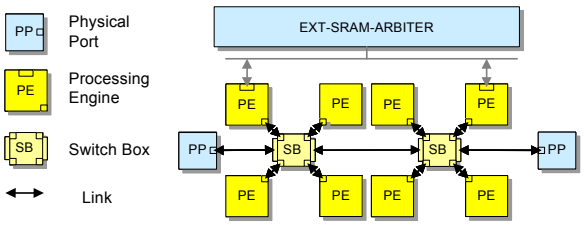

Figure 1: Block diagram of the multiprocessor

2. 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 [3] 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*.

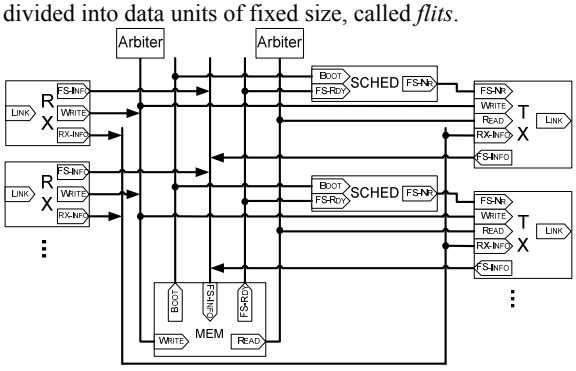

Figure 2: Block diagram of the switch box

Figure 2 shows the switch box (SB) of the NoC. The SB consists of receivers (RX), transmitters (TX), schedulers (SCHED), and flit-memory. Receiver and transmitter write/read flits to/from a shared memory (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:

$$1 \le EC_{SB,rx} \le N_{SB}$$

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 replies 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:

$$3 \le EC_{SB,tx} \le N_{SB} + 3$$

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 3 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.

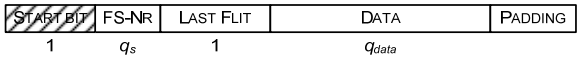

Figure 3: Segmented flit for off-chip communication

3. The processing engine

Figure 4 shows a block diagram of the processing engine that is used in our MPSoC. Central component of the proposed architecture is the S-Core-processor [1], 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 transmitting and receiving a flit is given by:

$$q_{data}/32+4 \le EC_{DL} \le q_{data}/32+6 \quad \text{and}$$

$$q_{data}/32+6 \le EC_{UL} \le q_{data}/32+8$$

To determine the resource efficiency of the software and of the hardware implementation, our VHDL-based characterisation environment PERFMON is used [3]. PERFMON provides an infrastructure for simulation and evaluation of the whole system, including main memory, debugging units, and performance counters. Each softcore processor can be used to substitute the currently used S-Core and to analyze its performance in the proposed multiprocessor environment. Because all parts of the PERFMON are synthesizable, the whole system can be mapped to a hardware technology, e.g., to an FPGA, for rapid prototyping.

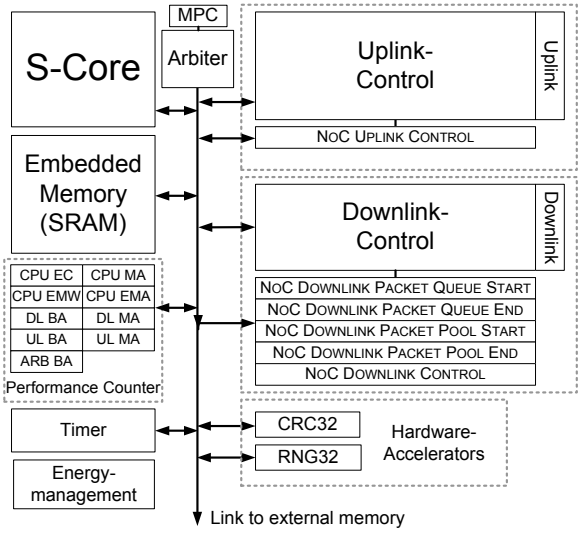

Figure 4: Block diagram of the processing engine

4. Test-Environment

Initial implementations of the system have been mapped to FPGA architectures and have been tested in our rapid prototyping environment RAPTOR2000. This allows for fast simulation and verification in early design stages. After successful testing, the FPGA prototype has been replaced by an ASIC implementation (see Figure 5). A daughter-board for RAPTOR2000 has been developed, comprising the MPSoC, 4 MB external memory, and a Spartan XC3S1500 FPGA, integrating an interface to the RAPTOR2000 motherboard. The user can easily interact with the MPSoC, using the PCI bus interface of RAPTOR2000.

As a proof-of-concept, the multiprocessor is integrated in the SAHNE simulation environment [2], 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 proposed MPSoC ASIC realization. The hardware is connected to the simulator using the hardware abstraction layer (HAL) of the packet processing library, which has been presented in [3]. This enables hardware/software co-simulation of large mobile ad-hoc-networks.

The multiprocessor has been manufactured in 180 nm UMC standard-cell-technology and occupies an area of 25 mm². At a clock frequency of 100 MHz, the average power consumption is 996 mW. At this speed, a communication bandwidth of up to 2.1 Gbps is achieved for each link of the NoC.

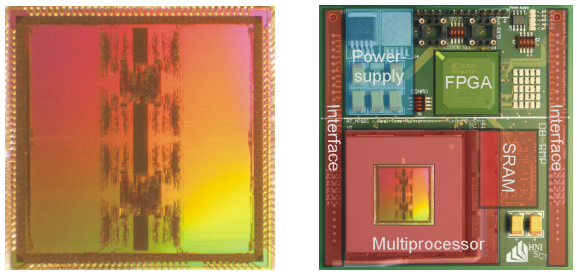

Figure 5 : Chip photo and test environment for the MPSoC

5. References

[1] D. Langen, J.-C. Niemann, M. Porrmann,, H. Kalte, and U. Rückert, "Implementation of a RISC Processor for SoC Design – FPGA Prototype vs. ASIC Implementation", *In Proc. of the IEEE-Workshop: Heterogeneous reconfigure-able Systems on Chip (SoC)*, Hamburg, Germany, 2002.

[2] K. Volbert, „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, 2002.

[3] Grünewald et al., "A Framework for Design Space Exploration of Resource Efficient Network Processing on Multiprocessor SoCs", *Crowley, P. and Franklin, M. A. and Hadimioglu, H. and Onufryk, P. Z., 12, pp. 245-277, Morgan Kaufmann Publishers,* 2005

Acknowledgement

The research described in this paper was funded by the Federal Ministry of Education and Research (Bundesministerium für Bildung und Forschung), registered there under grant number 01AK065F (NGN-PlaNetS), and by the DFG (Deutsche Forschungsgemeinschaft)