
SCALABLE SOFTWARE HARDWARE ARCHITECTURE PLATFORM FOR EMBEDDED SYSTEMS

Abstract

In order to handle the ever increasing complexity of applications found in modern embedded systems, the focus is moving away from single processor implementations towards heterogeneous multiprocessor system-on-chip (MPSoC) architectures. While offering high scale integration, high computing power and low power consumption, MPSoC designers also face new challenges. In particular, a mapping phase is required which cannot be found in single processor design but is essential to exploit the performance potentials of MPSoC architectures.

We are developing a framework, termed distributed operation layer (DOL), which seamlessly integrates the mapping of real-time applications onto MPSoC architectures into a complete design flow. The focus is on the mapping optimization that together with appropriate performance evaluation strategies allows to compute mappings with a guaranteed performance. Besides the mapping optimization, our framework provides models for application programmers, enabling them to write parallel applications that efficiently execute on MPSoC architectures.

Section

Computer Engineering Group

Head

Prof. Dr. Lothar Thiele

TIK Participants

Dr. Iuliana Bacivarov

Wolfgang Haid

Kai Huang

Funding

Sixth Framework Programme for
Research & Technological Development (FP6)
of the European Union

Web

www.shapes-p.org

www.tik.ee.ethz.ch/~shapes

SCALABLE SOFTWARE HARDWARE ARCHITECTURE PLATFORM FOR EMBEDDED SYSTEMS

Scope

Due to physical limitations, the performance increase of single processor systems will severely slow down in the near future. As CMOS technologies are continuously advancing, multiprocessor systems will be the only alternative to effectively use these advances and maintain the performance increase we have seen in the past 30 years.

Unfortunately, traditional methods from HW/SW codesign or the general purpose computing domain are not well suited to bring the performance potentials of MPSoC architectures to bear. The objective is therefore the design of scalable, programmable MPSoC architectures and to develop methods that allow the easy development of applications for these architectures. Our focus is on embedded real-time systems which necessitates a design process that allows continuous monitoring of quality of service requirements and guarantees the real-time constraints in a systematic manner.

In particular, to enable the execution of an application on a hardware platform, a software framework is required, see Fig. 1. In this context, we are interested in the mapping optimization and performance analysis at a high level of abstraction. Our goal is to develop the top-level software layer, namely the distributed operation layer (DOL), which allows to perform mapping optimization and performance analysis early in the design flow. The DOL is part of the SHAPES project (scalable software hardware architecture platform for embedded systems) in which a complete MPSoC hardware platform and the according design flow are developed.

Next, after giving a short overview over the applications, the software framework, and the hardware platform, we will present our main contributions, the technical approach, and potential impacts.

Applications

Ease of programming while maintaining high performance is paramount for the success of any hardware platform. With respect to this criterion, mainly two aspects need to be taken into account when developing a multiprocessor platform. On the one hand, knowledge about typical requirements and constraints of applications is needed. On the other hand, knowledge about the needs of the application programmers is needed.

The applications we are considering are real-time stream data processing applications, which need to be executed highly parallelized to meet the throughput and timing demands. In particular, we focus on applications from audio processing, medical imaging, and numerical simulation.

Software Framework

The objective of the software framework is to obtain the efficient execution of applications on the hardware platform. At the system level, this means that a large number of tasks and their communication need to be mapped efficiently onto the resources available on the hardware platform. At the resource level, this means that an efficient runtime environment consisting of the hardware dependent software and an operating system, as well as compilers need to be provided. To support the hardware design, to enable fast prototyping of applications, and to assess application performance, an instruction-accurate simulation framework is used.

The entire software framework is aware of the hardware architecture and its performance parameters. Incorporation of this knowledge in early design stages enables the fast exploration of the entire design space. This is the key to short design times which is one of the success factors in embedded system design where time-to-market pressure is high. On the other hand, only fast exploration capabilities guarantee a scalable software development process.

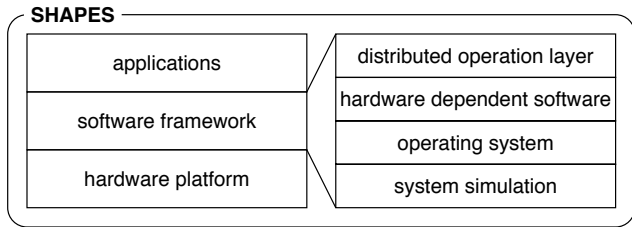


Fig. 1: Overview over the SHAPES MPSoC design

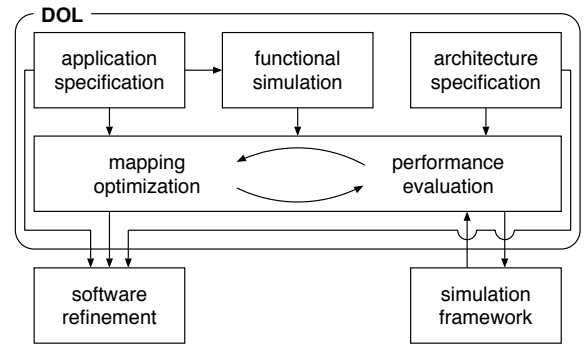


Fig. 2: Overview over the distributed operation layer

Hardware Platform

The SHAPES architecture is a tiled architecture that employs building blocks that are scalable on future CMOS technologies. A typical tile is composed of a VLIW DSP, a RISC processor, a distributed network processor, on-tile memories and a set of on-tile peripherals. Each tile can be equipped with a distributed external memory. The tile is the evolution of the Atmel Diopsis 740, a dual CPU processor integrating a floating-point VLIW mAgic DSP and an ARM7 RISC microcontroller unit. The SHAPES routing fabric connects on-chip and off-chip tiles, weaving a distributed packet switching network. Next-neighbor toroidal network topologies will be adopted for off-chip networking and maximum system density.

The considered architecture scales from low-end single modules hosting 32 tiles for mass market applications, to classic digital signal processing systems like radar and medical equipments (2000 tiles), and to high-end systems requiring massive numerical computation (32000 tiles).

Distributed Operation Layer

Located at the system level, the task of the distributed operation layer (DOL) is to generate an optimal mapping of a parallelized application onto a multiprocessor architecture in an automated manner. In particular, we have identified the following criteria which need special attention to reach this goal: parallelization, software refinement, scalability, design space exploration, and performance analysis.

The DOL can be considered as a framework tailored towards these criteria that actually implements the mapping optimization. It consists of basically four parts, as shown in Fig. 2. First, the DOL defines how to specify and program applications which is the basis for the automation of the software flow. Applications written according to this definition are amenable for automated refinement with respect to the hardware and the operating system. Second, the DOL defines how to specify the (abstracted) multiprocessor hardware architecture. Third, a purely functional simulation can be automatically generated based on the application specification. This simulation can be used for debugging and testing as well as for obtaining mapping relevant parameters at the application level. Fourth, the DOL implements a tool for mapping optimization which itself relies on a tightly coupled tool for performance analysis. Besides the high-level functional simulation, an external low-level simulation framework is loosely coupled with the mapping optimization tool for obtaining performance data.

The output of the DOL is a mapping specification which is used together with the application and architecture description as the input for further software refinements.

Technical Approach

To achieve our goals, we use the process network model of computation for specifying applications. This way, application programmers are able to expose an application's parallelism to the software framework. On the other hand, the process network semantics allows leveraging several structural properties during the software refinement, leading to a scalable design process.

Our main focus is on the mapping optimization and the performance analysis. We will apply multiobjective evolutionary algorithms for the former and formal performance analysis methods for the latter. In both areas, we can build on a sound foundation established in our group over several years.

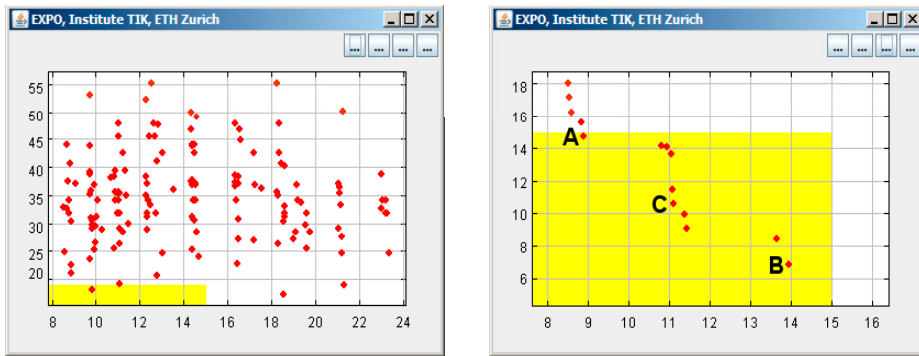


Fig. 3: Result of a multi-objective mapping exploration, starting from an initial population (left) towards an optimized population (right) with two objectives related to communication and computation

Potential Impact

In the development of the distributed operation layer, we are looking at multiprocessor embedded system design from different angles, keeping in mind the complete MPSoC design flow. Thereby, we are able to anticipate potential challenges and identify open questions in early design phases. On the other hand, applying our methods in the SHAPES project allows us to immediately assess the proposed solutions.

In particular, we apply our knowledge in system optimization and performance analysis to real-world problems which allows us to better understand the gap between the theoretic results and practical applications, opening the possibility for further improvement. One of these problems is that current optimization methods for mapping of large applications to multiprocessor systems are insufficient from the aspect of scalability. Also, formal performance analysis is currently only possible for a very restricted class of systems. We are convinced that we can make valuable contributions in solving these problems.

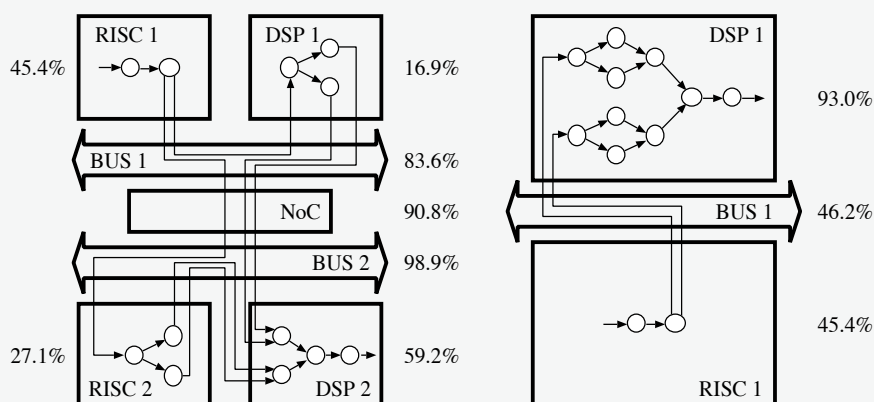


Fig. 4: Two sample architectures corresponding to points A and B from the population shown in Fig. 3

Related Publications

W. Haid, L. Thiele

Complex Task Activation Schemes in System Level Performance Analysis

5th Int'l Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'07), pages 173–178, Salzburg, Austria, Oct. 2007

L. Thiele, I. Bacivarov, W. Haid, K. Huang

Mapping Applications to Tiled Multiprocessor Embedded Systems

7th Int'l Conference on Application of Concurrency to System Design (ACSD'07), pages 29–40, Bratislava, Slovak Republic, Jul. 2007

K. Huang, L. Thiele, T. Stefanov, E. Deprettere

Performance Analysis of Multimedia Applications using Correlated Streams

Design, Automation and Test in Europe (DATE'07), Nice, France, pages 912–917, Apr. 2007

P. S. Paolucci, A. A. Jerraya, R. Leupers,

L. Thiele, P. Vicini

SHAPES – A Tiled Scalable Software Hardware Architecture Platform for Embedded Systems

4th Int'l Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS'06), Seoul, Korea, pages 167–172, Oct. 2006

E. Zitzler, L. Thiele

Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach

IEEE Transactions on Evolutionary Computation, Vol. 3, No. 4, pages 257–271, Nov. 1999

J. Teich, T. Blickle, L. Thiele

System-Level Synthesis Using Evolutionary Algorithms

J. Design Automation for Embedded Systems, Vol. 3, pages 23–58, Jan. 1998