Tutorial

MNEMEE – A Framework for Memory Management and Optimization of Static and Dynamic Data in MPSoCs

Organizer:
Dr. Arindam Mallik
IMEC vzw
Leuven, Belgium 3000
arindam.mallik@imec.be

Prof. Peter Marwedel
TU Dortmund, Germany

Prof. Dimitrios Soudris
National Technical University of Athens, Greece SA

Dr. Sander Stuijk
Eindhoven University of Technology, The Netherlands

TUTORIAL ABSTRACT

As embedded systems are becoming the center of our digital life, system design becomes progressively harder. The integration of multiple features on devices with limited resources requires careful and exhaustive exploration of the design search space in order to efficiently map modern applications to an embedded multi-processor platform. The MNEMEE project [1] addresses this challenge by offering a unique integrated tool flow that performs source-to-source transformations to automatically optimize the original source code and map it on the target platform. The optimizations aim at reducing the number of memory accesses and the required memory storage of both dynamically and statically allocated data. Furthermore, the MNEMEE tool flow performs optimal assignment of all data on the memory hierarchy of the target platform. Overall, the MNEMEE techniques embedded in it will lead to more cost efficient systems that offer a better performance and lower energy consumption.

This tutorial gives an overview of the MNEMEE tool flow. The objective of the tutorial is to familiarize the audience with the tool framework and the optimizations used in the individual tools. The tutorial also features a demonstration of the tool flow. This demonstration shows that the tools developed in the MNEMEE project provide a user-friendly and efficient framework for MPSoC programming and memory management.

Categories & Subject Descriptors: C.3 [Special-Purpose and Application-Based Systems]: [Real-time and embedded systems, Signal processing systems]

General Terms: Design, Performance, Experimentation.

Keywords: MPSoc Design, Embedded software, Memory Management, Automatic Parallelization

MNEMEE Toolflow

The MNEMEE tool flow provides a completely automated trajectory to map sequential applications onto a MPSoC while exploiting its memory hierarchy.

CASES'10, October 24–29, 2010, Scottsdale, Arizona, USA. ACM 978-1-60558-903-9/10/10.

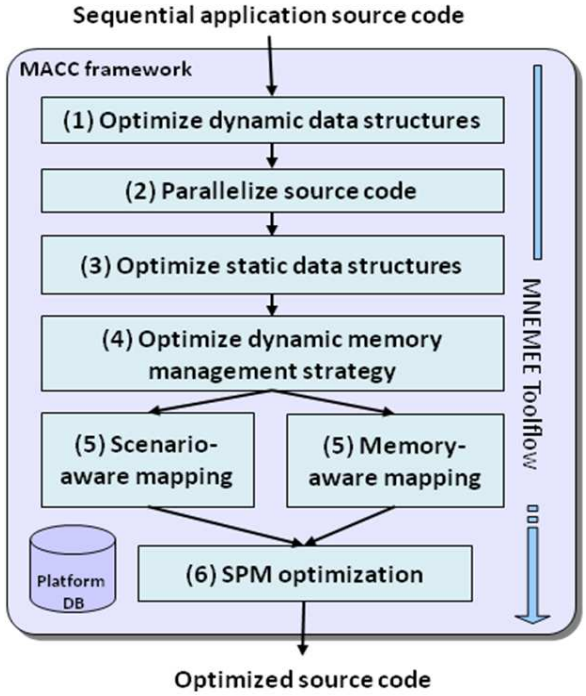

Figure 1. MNEMEE Toolflow

Figure 1 demonstrates the MNEMEE tool flow. The input of the tool flow is sequential source code written in C. The output of the flow is parallelized source code that is optimized for the target MPSoC and its memory hierarchy. The first step optimizes dynamically allocated data structures by changing their implementation [2]. It is followed by Step 2 which identifies any potential parallelization in the source code and implements it by breaking the code into several parallel tasks [3]. The statically allocated arrays are optimized and allocated in the memory hierarchy in step 3 [4]. Since the parallelization of the original source code has already taken place, the tool can map the statically allocated data structures efficiently onto the memory hierarchy, as their size and behavior is known. The dynamically allocated data structures are handled in the next step, and mapped onto the memory hierarchy [4]. Step 5 maps the parallelized application

onto the processors and memories. The MNEMEE tool flow offers two alternatives mapping techniques. The first technique called scenario-aware mapping [5], tries to exploit the dynamic behavior of an application in order to save resources. The second technique, called memory-aware mapping, focuses on finding a mapping that minimizes the energy consumption of the memory subsystem. Finally, step 6 further optimizes the scratchpad allocation of each processor in the target platform. The final parallelized application is a set of parallel C source files that collectively represents the same functionality of the sequential input version.

To combine the large number of required processing steps into a single tool flow, the MACC framework for source level optimization development has been used. This framework exploits the abstract syntax tree code representation provided by the ICD-C compiler development framework. MACC provides a common platform model along with a well defined interface for integration of analysis and optimization tools. Furthermore, a graphical user interface is provided to enhance the usability of the tool flow.

Tutorial Description

This tutorial will focus on the design challenges that exist in the field of MPSoC system design. It will discuss the optimizations and analysis a system designer needs to perform for an effective and intelligent mapping of multimedia and communication application on an MPSoC platform. The speakers will present the various design level optimizations and the systematic methodology that were developed in the MNEMEE project. They will also discuss the set of integrated system level tools that implement these optimization techniques and methodologies. These tools provide an easy-to-use approach for embedded designers to map and optimize embedded applications for MPSoC platforms. The speakers will also show initial results that demonstrate the benefits of the MNEMEE tools and methodologies.

The tutorial will consist of the following sessions:

- Introduction to MPSoC design and the MACC framework
- Automatic application parallelization techniques for MPSoCs
- Static and dynamic memory optimization methodologies for MPSoCs
- Methodologies to map applications onto MPSoCs
- Structured methodologies to integrate different optimization and analysis tools for MPSoC system design in a single framework
- A live demonstration of the MNEMEE tool

Biography of the Speakers

Dr. Arindam Mallik received his Masters and Ph.D. degree in Electrical and Computer Engineering from Northwestern University (Evanston, USA) in 2004 and 2008, respectively. Currently he works as a specialist researcher in Interuniversitair Micro-Elektronica Centrum vzw (imec) in the field of embedded systems and system level optimization of novel memory technology. He has published more than 20 papers in International Journals and Conferences and holds two patents. His research interests include adaptive system architecture, power and performance tradeoff analysis in embedded platforms, and dynamic frequency and power management in mobile platforms. Arindam acts as the project coordinator for the MNEMEE project.

Prof. Peter Marwedel received his Ph.D. in Physics from the University of Kiel (Germany) in 1974. He is a pioneer of behavioural synthesis and worked on the MIMOLA synthesis and code generation tools (a landmark in this area) until 1986. In 1987, he received the Dr. habil. degree (a degree required for tenure track positions in Germany) for this work. Since 1989 he is a Full Professor at the Computer Science Department of the University of Dortmund (Germany). Dr. Marwedel is a member of the ARTIST European network of excellence on real-time and embedded systems and he published a very influential text book on embedded system design. In 2003, he received the teaching award of his University.

Prof. Dimitrios Soudris received his Diploma in Electrical Engineering from the University of Patras, Greece, in 1987. He received the Ph.D. Degree in Electrical Engineering, from the University of Patras in 1992. He is currently working as Assistant Professor in School of Electrical and Computer Engineering of National Technical University of Athens, Greece. Also, he is a member of Institute of Communication and Computer Systems. He has published more than 210 papers in international journals and conferences. Additionally, he is co-editor in four books of Kluwer and Springer. He is project leader and principal investigator in numerous research projects funded by the Greek Government and Industry as well as the European Commission. Also, he received an award from INTEL and IBM for the project results of LPGD.

Dr. Sander Stuijk received his Master's degree in Electrical Engineering in 2002 and his Ph.D. degree in 2007 from the Eindhoven University of Technology. He is currently an assistant professor in the Department of Electrical Engineering at the same university. He has been involved with different European and national research projects. His research interests include modeling and mapping techniques for the design, specification, analysis and synthesis of predictable hardware/software systems.

Daniel Cordes is a doctoral student working with Prof. Peter Marwedel. He has reached his diploma degree in 2008 by focusing on static analysis problems. His current research topic is the development of methods for automatic translation of sequential C-Code into an optimized, parallel version.

References

[1] The MNEMEE project webpage, www.mnemee.org

[2] C. Baloukas et al., “Optimization methodology of dynamic data structures based on genetic algorithms for multimedia embedded systems,” in Journal of Systems and Software, Volume 82, Issue 4, April 2009, Pages 590-602

[3] D. Cordes et al., “Automatic Parallelization of Embedded Software Using Hierarchical Task Graphs and Integer Linear Programming ”, in the proceedings of The International Conference on Hardware-Software Codesign and System Synthesis (CODES+ISSS), October 2010

[4] Y. Iosifidis et al, “A Framework for Automatic Parallelization, Static and Dynamic Memory Optimization in MPSoC Platforms”, in the Design Automation Conference (DAC-2010), June 2010

[5] S. Stuijk et al. “A Predictable Multiprocessor Design Flow for Streaming Applications with Dynamic Behaviour”, In the proceedings of the Conference on Digital System Design (DSD), September 2010