MNEMEE – An Automated Toolflow for Parallelization and Memory Management in MPSoC Platforms

Arindam Mallik[1], Stylianos Mamagkakis[1], Christos Baloukas[2], Lazaros Papadopoulos[2], Dimitrios Soudris[2], Sander Stuijk[3], Olivera Jovanovic[4], Florian Schmoll[4], Daniel Cordes[5], Robert Pyka[5], Peter Marwedel[5], François Capman[6], Séverin Collet[6], Nikolaos Mitas[7], Dimitrios Kritharidis[7]

[1]IMEC vzw, Belgium (arindam, mamagka)@imec.be; [2]Institute of Communication and Computer Systems, Greece (dsoudris@microlab.ntua.gr, cmpalouk@ee.duth.gr); [3]Eindhoven University of Technology, Netherlands (s.stuijk@tue.nl); [4]TU-Dortmund, Germany (florian.schmoll@tu-dortmund.de, olivera.jovanovic@udo.edu); [5]Informatik Centrum Dortmund, Germany (marwedel, pyka, cordes)@icd.de ; [6]THALES Communications, France (francois.capman, severin.collet)@fr.thalesgroup.com; [7]INTRACOM Telecom, Greece (nmitas, dkri)@intracom.gr

1. INTRODUCTION

Mobile, intelligent devices that are able to deliver communication services and multimedia content anytime, anywhere are the dominant players in the field of embedded systems. These systems combine many different streaming applications (e.g., H.264-AVC, JPEG2000, WiMax) in a single system. The basic characteristic of these applications are typically large computational requirements and intensive data transfer and storage needs. As a result, the primary platform for such applications is Multiprocessor Systems-on-Chip (MPSoCs). These MPSoCs can deliver the computational power required by novel applications. Modern MPSoCs contain a complex memory hierarchy that allows applications to meet their data transfer and storage requirements. However, it brings the additional challenge to the system designers to efficiently map applications onto processors and memories. The design choices have a large impact on the energy consumption and memory footprint of the final system. This in the end has a direct impact on the system cost and the battery lifetime of the system, i.e., the user experience.

The basic constraints for MPSoC mapping can be categorized as follows:

- Parallelization of the sequential application code.
- Optimization of the static and dynamic data structures for efficient and intelligent utilization of the memory hierarchy.
- Mapping of the parallelized source code onto the computational and storage resources available in the MPSoC platform.

The MNEMEE project [1] addresses the aforementioned challenges by introducing a novel tool flow that integrates several state-of-the-art source-to-source optimization methodologies and tools. It provides a methodology to automatically parallelize the source code of an application. It also optimizes the static and dynamic data structures in the source code such that they can efficiently use the memory hierarchy in an MPSoC. Finally, the tool flow maps the parallelized source code onto the processors and memories in an MPSoC. Many of the methodologies that are used in the tool flow are based on multi-objective exploration strategies. This allows designers to make design trade-offs and it makes product customization at design-time much easier. The MNEMEE tool flow provides a completely automated trajectory to map sequential applications onto an MPSoC while exploiting its memory hierarchy. The primary objective is to reduce the energy consumption and design-time of the new embedded system.

2. MNEMEE TOOLFLOW

The MNEMEE tool flow is shown in Figure 1. The flow takes sequential C source code of an application as input. In several steps it parallelizes and optimizes this C code for implementation onto an MPSoC.

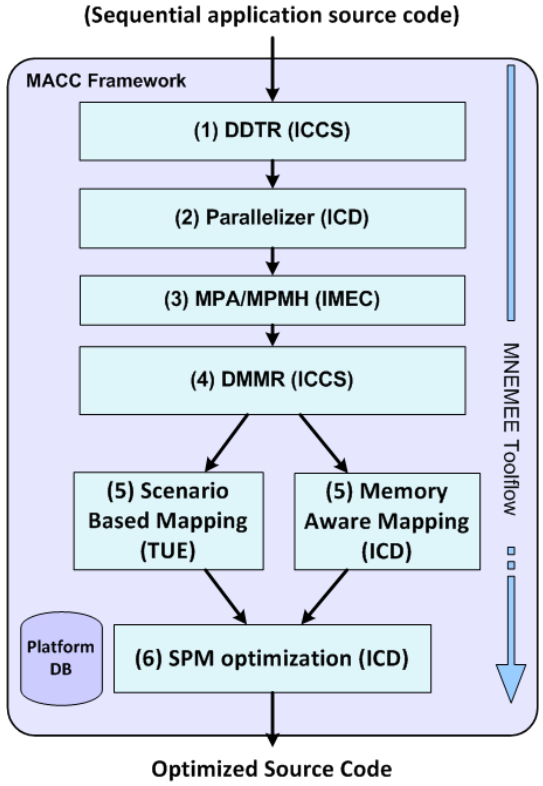

Figure 1 - MNEMEE tool flow

The first step optimizes, based on the access pattern, the dynamic data structures in the original source code. It changes the implementation of all dynamic data structures like dynamic arrays, linked lists and trees, based on the Dynamic Data Type Refinement (DDTR) methodology [2]. The parallelization tool [3] receives as input the sequential source code that has already been optimized in step 1. Step 2 must decide how the sequential application source code will be parallelized. It uses an integer linear programming (ILP) based approach to partition the application into several concurrently executed tasks. Step 3 implements this parallelization, i.e., it transforms the source code. The result of the optimization is a transformed parallelized

application with data copies and block transfers explicitly expressed in the source code, automatic handling of synchronization of data, and a mapping of the data and copies to the various memory layers [4]. Step 4 performs a similar optimization on the dynamic data structures. The next step maps the parallelized application onto the processors and memories in the MPSoC. For the mapping step, the toolflow offers two alternatives. The user can exploit the dynamic behavior of an application in order to save resources using the scenario-aware mapping techniques [5]. Alternatively, the main objective for mapping could be to optimize the energy consumption of the memory subsystem by opting for the memory-aware mapping. The scratchpad memory allocation tool (step 6) is the last step in the MNEMEE tool flow. It finalizes the mapping decisions for the data objects that are used in the application. Earlier tools have provided hints on the mapping of the data structures onto the memories. The step can take full advantage of all mapping decisions and optimizations performed in earlier steps of the flow.

The tool flow combines a large number of different tools in a single framework. To achieve the required level of integration, the tool flow has been build using an existing compiler development framework, ICD-C, in combination with the MACC framework. The latter framework offers a structured method to integrate different optimization and analysis tools. This framework was partially developed in this project [6]. Furthermore, a graphical user interface was developed to enhance the usability of the tool flow.

Since every step of the tool flow is fully automated, it enables the designer to perform an optimal non-overlaying memory allocation with almost no effort while significantly reducing the design time for embedded systems.

3. USE CASES FOR INDUSTRY

The evaluation of an automated toolflow can only be successfully evaluated by actual use of it in an industrial setup. For the MNEMEE toolflow, it has been evaluated through the development of two industrial partners for different target application domains, namely multimedia and communications. Both examples demonstrate the integration of the MNEMEE tools into their design flow, emphasizing the automation achieved.

Thales Communications, France (TCF)'s targeted application is based on NATO standard STANAG 4591 implementing the enhanced Mixed Excitation Linear Predictive (MELPe) algorithm at 2400, 1200 and 600 bit/sec. The speech signal is sampled at 8 kHz. At 2400bits/s, the frame length of the input signal is 22.5 ms, where at 1200 bits/sec and 600 bits/sec three and four consecutive frames (respectively) are grouped into a super-frame. The resulting super-frame is jointly quantized to obtain high coding efficiency. The targeted functionality is an integrated real-time solution of the different modes allowing dynamic switches between the different rates on the selected hardware platform. The target platform is an OMAP L137 provided by Texas Instrument which is a heterogeneous MPSoC with an ARM processor and a DSP.

In the framework of MNEMEE, Intracom Telecom (ICOM) targets the IEEE 802.16e system, widely known as Mobile WiMAX. IEEE 802.16e is a broadband wireless solution that enables convergence of mobile and fixed broadband networks through a common wide-area broadband radio access technology and flexible network architecture. With a fast air link, its asymmetric downlink/uplink capability, a fine resource granularity and a flexible resource allocation mechanism, Mobile WiMAX is designed to meet QoS requirements for a wide range of data services and applications. The selected platform for Intracom is MSC8144 processor, a high-performance multicore DSP from Freescale that targets wireline and wireless infrastructure applications. This multicore DSP combines four fully-programmable StarCore™ DSP cores, each running at up to 1 GHz with an architecture highly optimized for voice, fax, video, and data compression processing. An internal QUICC Engine™ dual-RISC packet-processor supports multiple networking protocols to guarantee reliable data transport over packet networks while significantly offloading such processing from the DSP cores.

Henceforth, both homogeneous and heterogeneous MPSoC platforms were used for the MNEMEE toolflow evaluation. Also, the different nature of application domain helped to look at both functional and data-level parallelization.

3.1 Experimental Results

For TCF, optimization tools as designed and developed within MNEMEE project are of great importance in order to fully benefit from emerging MPSoC architectures. If it remains a challenge to provide a fully automated optimization tool flow with optimal performance, the results obtained within MNEMEE demonstrate that a significant step has been achieved. In the case of ICOM, participation in the MNEMEE project enabled the company to have early access to innovative technology that can have a real impact in the design of contemporary complex embedded systems. The benefits offered by the tools have been evaluated using the platform described earlier. In some cases, designers had to use analytical models to estimate energy consumption as those numbers were not directly available from the platform. The evaluation showed significant gains in design time, memory bandwidth, memory footprint, and energy consumption as summarized in Table 1. The benefits observed were dependent on the applications as well as the platform used. The details of the results are available in the project homepage [1].

Table 1: Summary of results for use-cases

	TCF	ICOM
Memory Footprint	30 %	~1%
Memory Bandwidth	No gain	17%
Energy Consumption	52 %	Not measurable
Design Time	76 %	38%

4. REFERENCES

[1] MNEMEE Project (IST-216224) - http://www.mnemee.org/.

[2] C. Baloukas et al., "Optimization methodology of dynamic data structures based on genetic algorithms for multimedia embedded systems," in Journal System Software, 82, 4 (April 2009), p. 590-602.

[3] D. Cordes, P. Marwedel and A. Mallik. Automatic Parallelization of Embedded Software Using Hierarchical Task Graphs and Integer Linear Programming, CODES+ISSS'2010, October 2010

[4] Y. Iosifidis, A. Mallik, et al., "A Framework for Automatic Parallelization, Static and Dynamic Memory Optimization in MPSoC Platforms" in the Design Automation Conference, DAC 10, June 2010

[5] Stuijk, M.C.W. Geilen and T. Basten, "A Predictable Multiprocessor Design Flow for Streaming Applications with Dynamic Behaviour" in DSD 10, September 2010.

[6] R. Pyka, F. Klein, et al. "Versatile System-level Memory-aware Platform Description Approach for embedded MPSoCs", LCTES 2010, April 2010