SMECY: Smart Multi-core Embedded SYstems

[Special Session]

Francois Pacull
CEA-Leti Minatec,
Grenoble, France
francois.pacull@cea.fr

Koen Bertels
Delft University of Technology,
Delft, Netherlands
k.l.m.bertels@tudelft.nl

Martin Danek
Department of Signal
Processing, UTIA,
Praha, Czech Republic
danek@utia.cas.cz

Giulio Urlini
ST Microelectronics,
Milan, Italy
giulio.urlini@st.com

ABSTRACT

SMECY project is an ambitious European initiative involving 29 partners across 9 countries to enable Europe to have a leader role in multi-core domain by developing new programming technologies enabling the exploitation of architectures offering hundreds of cores. Multi-core technologies will rapidly provide to the parallel computing field improved performance, energy saving and cost reduction and will become of strategic value in winning market share in all areas of embedded systems. Given the need, SMECY lays the focus on targeting programming multi-core architecture for consumer electronics with efficient resources management. The first presentation describes the overall project while the two others are respectively dedicated to the multi-core platforms targeted in the project and the description of the tools constituting the bricks of the tool chains.

Categories and Subject Descriptors

D.2 [**SOFTWARE ENGINEERING**]: General
; D.2.6 [**Programming Environments**]: Integrated environments

General Terms

Theory

Keywords

compilation chain, multi-core

1. INTRODUCTION

For years the research community tried to find the grail in targeting a unique chain of tools which would work efficiently for any platform, any type of parallelism and any application domain. It is time to realize that such a ”one fits all” chain does not exists and the increase of number of the computing units will not make the task easier. The conceptual approach of SMECY is based on the statement that a tool chain should take both the application requirements and the platform specificities of various embedded systems in different industries into account to become efficient.

Three presentations covering the different aspects of the project are proposed. The first describes the overall project while the two others are respectively dedicated to the multi-core platforms targeted in the project and the description of the tools constituting the bricks of the tool chains.

2. PROGRAMMING TOOL CHAINS FOR MULTI-CORE PLATFORMS

In SMECY we consider that i) once one type of parallelism and one hardware architecture is fixed it is possible to provide an efficient chain of compilation with appropriate tools to analyse and optimize the execution; ii) once this set of tools is available, it is possible at reasonable cost to go from one hardware architecture to another one; iii) alternatively, it is possible to go from one type of parallelism to another. This first presentation describes the work planned in the project and the expected results.

The applications providers of SMECY consortium have selected more than 15 applications that can benefit from these new multi-core platforms. These applications have been clustered into the following three sets: *Radar signal processing and earth observation, Multimedia, mobile and wireless transmission and Stream processing (e.g. video surveillance).* These three clusters define three types of parallelism that could take advantage of computing power offered by the 2 SMECY multi-core platforms. These plaforms are presented in details in the presentation *The SMECY multi-core platforms* by Giulio Urlini and Martin Danek (See section 3).

The project would be useless without the contribution of several tools providers that will act at different stages in the tool chain from the application description (programming model) to the execution code running on the different cores of the platforms. This includes tools analysing the application and the platform characteristics in order the provide the best deployment according to several parameters such as the performance, the power cost or the management of thermal aspects. This is detailed in the presentation *The SMECY tool chain : customization in view of application and platform characteristics* by Koen Bertels (See section 4).

GLSVLSI'11, May 2–4, 2011, Lausanne, Switzerland.
ACM 978-1-4503-0667-6/11/05.

3. SMECY MULTI-CORE PLATFORMS

The aim of the SMECY project is to provide tool chains for programming multi-core architectures. Two platforms are targeted for their radically different approaches: P2012 is provided by an industrial partner ST and EdkDSP by an academic partner UTIA.

The EdkDSP platform is based on the classical master-worker architecture. The worker is represented by a Basic Computing Element (BCE) that is connected to a master represented by a central 32-bit microprocessor MicroBlaze that acts as an end-user interface and performs task allocation and data transfers. The BCE consists of a simple CPU (sCPU, in our case 8-bit PicoBlaze) in the role of a programmable finite state machine that controls a configurable Data-Flow Unit (DFU) implementing in hardware a number of computational kernels. The BCE is generic. The system designer can specify basic functions calculated in the dataflow unit to suit a given application domain (image or audio processing). Since the internal organization of the BCE decouples control flow from the data flow, it enables polymorphism in cases where the same flow control can be used in different data domains (e.g. different floating-point precision). Moreover, it enables to use DFUs of different complexity in terms of implemented operations with different resource requirements (silicon area, power, graceful degradation in case of faults, etc.) since it is assumed that more specialized operations implemented directly in a complex DFU can be realized using a sequence of generic operations in a simpler DFU. Each case is represented by a different sCPU firmware subprogram: the subprogram for the more complex DFU will contain only a single instruction, while the one for the simpler DFU will contain an equivalent sequence of instructions.

The BCE worker has been designed so as to hide unnecessary implementation details from the designer. The central 32-bit microprocessor perceives a BCE as an intelligent memory without any direct access to the BCE internals. On the program level the BCE interfacing appears as a number of function calls in a high-level language with each function representing one complete computation in the BCE.

P2012 is an area- and power-efficient many-core computing fabric, and it provides an architectural harness that eases integration of hardwired accelerators.

The P2012 computing fabric is highly modular, as it is based on multiple clusters implemented with independent power and clock domains, enabling aggressive fine-grained power, reliability and variability management.

Clusters are connected via a high-performance fully-asynchronous network-on-chip (NoC), which provides scalable bandwidth, power efficiency and robust communication across different power and clock domains. Each cluster features up to 16 tightly-coupled processors sharing multi-banked level-1 instruction and data memories, a multi-channel advanced DMA engine, and specialized hardware for synchronization and scheduling acceleration. P2012 achieves extreme area and energy efficiency by aggressive exploitation of domain-specific acceleration at the processor and cluster level. 1 Each processor can be specialized at design time with modular extensions (vector units, floating-point unit, special-purpose instructions). Clusters can easily become heterogeneous computing engines thanks to the integration of coarse-grained hardware processing elements that implement those functions for which a software implementation would be inefficient (e.g. bitwise stream manipulations). Hardware-software interaction is facilitated by the local and global interconnect which efficiently supports point-to-point stream communication.

4. SMECY TOOL CHAINS: CUSTOMIZATION IN VIEW OF APPLICATION AND PLATFORM CHARACTERISTICS

Multi-core platforms, especially when having some heterogeneous components, have the reputation to be difficult to program. The Cell processor never really managed to change this perception resulting, among other reasons, ultimately in its retirement.

One of the challenges when trying to execute application on multi-core platforms is to map the application in an efficient way on the available hardware resources where the availability of those resources can also be depending on some runtime state. This assumes that the system developer has a good understanding of the features and resources of the targeted platform. This requirement however is orthogonal to efficient and portable system design. The reason is that the strategies used to map an application, implying certain code transformations, will differ in view of the assumed architectural template and the characteristics of the application. For each of the target architectures, a tool chain will be defined that takes both the application and the platform characterization as input parameters and determines the relevant transformation and mapping decisions. In both cases, the number of design decisions (such as different partitioning, code transformations and mapping choices) that need to be made outnumber what one can manually assess and explore. The SMECY tool chain will therefore focus on (semi) automatic tools support. The functionality offered will range from extraction of application and platform characteristics, the automatic transformation and partitioning and mapping up to the code generation for all the available hardware components.

Design time tools will have to be complemented by runtime functionality to provide e.g. appropriate scheduling and the capability to deal with faulty components. Runtime profiling, monitoring and scheduling will be investigated. We will present a detailed description of the prototypical tool chains that are expected towards the end of the project and how we intend to assess their usefulness on the two stated platforms for a wide range of applications domains.

5. CONCLUSION

The key outcomes of the SMECY project are programming and design methods, multi-core architectural solutions and associated supporting tools to master smart multi-core embedded systems in different application domains. This is an ambitious European initiative with the ultimate aim of enabling Europe to play a key role in the multi-core field.

Acknowledgment

The research leading to these results has received funding from the ARTEMIS Joint Undertaking under grant agreement number 100230 and from the national programmes / funding authorities.