

Rigorous System Level Modeling and Analysis of Mixed HW/SW Systems

P. Bourgos, A. Basu, M. Bozga, S. Bensalem, J. Sifakis
UJF-Grenoble 1 / CNRS, VERIMAG UMR 5104
Grenoble, F-38041, France
fbourgos, basu, bozga, bensalem, sifakis@imag.fr

K. Huang
Institute of VLSI Design
Zhejiang University, China
huangk@vlsi.zju.edu.cn

Abstract—A grand challenge in complex embedded systems design is developing methods and tools for modeling and analyzing the behavior of an application software running on multicore or distributed platforms. We propose a rigorous method and a tool chain that allows to obtain a faithful model representing the behavior of a mixed hardware/software system from a model of its application software and a model of its underlying hardware architecture. The system model can be simulated and analyzed for validation of both functional and extra-functional properties. The tool chain uses DOL (Distributed Operation Layer [1]) as the frontend for specifying the application software and hardware architecture, and BIP (Behavior Interaction Priority [2]) as the modeling and analysis framework. It is illustrated through the construction of system models of MJPEG and MPEG2 decoder applications running on MPARM, a multicore architecture.

I. INTRODUCTION

Performance of embedded applications strongly depends on features of the underlying hardware platform. In contrast to performance of application software running on a single core, getting the maximum throughput out of multicore processors demands application software to be designed taking parallelism into account from scratch. This is needed to catch up with the fast growth of computing capacity due to the foreseeable exponential increase of physical parallelism. But programming, testing and verifying parallel software with currently existing tools is notoriously hard, even for experts. There are no rigorous techniques for deriving global model of a given system from models of its application software and its execution platform.

Application software must be programmed for performance, in a platform independent way, exhibiting all potential parallelism. Its implementation must deal with mapping the specified application-level parallelism onto platform-level (threads, cores, processors) on an as-needed/as-available basis. Actually, this mapping would need to be adapted dynamically as applications must scale up or down

The research leading to these results has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under grant agreement no 248776 (PRO3D) and from ARTEMIS JU grant agreement ARTEMIS-2009-1-100230 (SMECY)

according to the available resources of the execution platform. Moreover, efficiency and correctness are not the only concerns. Programmer productivity, that is, the programmer's ability to design correct software that gathers the maximum performance out of an arbitrary multicore platform with ease should not be neglected [3].

Achieving these goals requires a design flow based on a single semantic model. The design flow must be able to generate rigorous models of mixed hardware/software systems, suitable for analysis, design space exploration and automatic code generation. The main contribution of this paper is deriving a rigorous system model combining the application software and the architecture, which can be the basis for multiple objectives, such as functional verification, performance evaluation and code generation for target architectures.

We propose a system construction method that is both rigorous and allows a fine analysis of system dynamics. It is rigorous because it is based on formal models, have precise semantics and thus can be analyzed by using formal techniques. A system model is derived by progressively integrating constraints induced on an application software model by the underlying hardware architecture model. Both models are described in BIP [2], which is a formal component based modeling framework. In contrast to ad hoc modeling approaches, the system model is obtained from a BIP model of the application software and a description of the hardware architecture, by application of source-to-source transformations that are correct-by-construction [4]. The final generated model is a mixed software-hardware model which provides the capability using a single model to simulate and apply formal verification techniques on it using the BIP framework.

Metro II [5] is a platform-based design framework and provides a simulation backend based on SystemC. Octopus [6] allows design space exploration by stochastic simulation of task graphs. Both have connections to formal verification tools based on model checking. Most of the frameworks for mixed HW/SW systems are based on SystemC [7] as a language for modeling at various levels of abstractions. Various tools and associated design methodologies emerged

