Enabling Timing and Power Aware Virtual Prototyping of HW/SW Systems

Kim Grütter, Kai Hylla, Sven Rosinger, Philipp A. Hartmann
OFFIS – Institute for Information Technology
Technology Cluster Design Hardware/Software Systems
Oldenburg, Germany
Email: {gruettner | hylla | rosinger | hartmann}@offis.de

Wolfgang Nebel
Carl von Ossietzky University Oldenburg
Embedded Hardware/Software Systems
Oldenburg, Germany
Email: nebel@informatik.uni-oldenburg.de

Abstract—We propose the concept of an ESL framework for timing and power aware rapid virtual system prototyping of embedded HW/SW systems. Our proposed flow combines system-level timing and power estimation techniques available in commercial tools with platform-based rapid prototyping. Our proposal aims at the generation of executable virtual prototypes from a functional C/C++ specification. These prototypes are enriched by static and dynamic power values as well as execution times. They allow a trade-off between different platforms, mapping alternatives, and optimization techniques, based on domain-specific workload scenarios.

I. INTRODUCTION

To capture the entire behaviour of today’s large and complex SoCs, the different components of the system cannot be considered separately. For analysis of the dynamic behaviour this implies a behavioral simulation of all system components and their interaction, which is essential for power and timing estimations. For specific platforms, proprietary simulation environments are available for both timing and power models. But a common and open framework, suitable for a wide range of platforms and designs is still missing. Such a framework would allow comparing different platform characteristics and thus enable rapid prototyping and design space exploration. Performance bottlenecks and power peaks within the entire system could be identified in early design phases, where modifications of the system are easier and more affordable than in later phases.

In this overview, we propose a new concept for a system-level estimation and simulation framework for timing and power aware analysis, that is currently under implementation by the COMPLEX FP7 European integrated project [1]. This framework follows a unified system-level specification for HW and SW, but utilizes different estimation techniques for custom HW and SW, as well as pre-defined IP components.

II. GOALS

Generally, embedded system design is a sequence of decisions finally leading to an implementation. These decisions have to consider the design expectations in terms of functional and non-functional system properties, which need to be captured and tracked during the entire design flow. All dimensions of the design space (e.g. functional, timing, power, performance, memory, area,...) need to be explored in order to estimate the cost of the final implementation. For this purpose a dependable and accurate model of the target system and workload scenario is necessary prior to heavy investments in HW and SW development. The focus in this work is the enrichment of virtual execution platforms with the aforementioned non-functional properties in early design phases.

Currently existing virtual platform simulation tools are expensive and suffer from the missing availability of a wide range of different platform models. Additionally, existing models do not offer enough flexibility in terms of design space exploration. Most virtual platforms are targeting mainly functional and loose timing aspects in early software development, which is of course important but not enough for the demanding workloads and tight non-functional constraints of tomorrow [2]. Code generation in Model-Driven Engineering (MDE), high-level synthesis techniques for custom HW, and abstraction of existing components to system-level descriptions are key enablers for unified system-level simulation approaches [3].

We want to consider functional, power and timing behaviour on system-level after manual partitioning and mapping to an implementation platform. This requires an executable model of the entire system, which is capable to simulate large sequences of domain-specific workloads in an acceptable time. This cannot be done at RT level due to high simulation complexity. To the best of our knowledge, there is no integrated framework available that enables the estimation and simulation of timing and power properties from a HW/SW independent executable specification model under manual mapping constraints. The goals of our presented approach are: (1) combination and augmentation of well established commercial ESL synthesis and analysis tools into a seamless design flow enabling performance and power aware virtual prototyping from a combined HW/SW perspective; (2) platform vendor and application domain independent framework with open interfaces; (3) fast simulation and assessment of functional and non-functional (time and power) properties of the entire system after platform mapping with scalable accuracy; (4) and interface to model-driven SW design entry using MARTE/UML and the industry standard Matlab/Simulink model-based design environment.
III. PROPOSED CONCEPT

The proposed concept for a rapid prototyping framework is illustrated in Fig. 1. Following the platform-based-design approach [4], the executable specification is separated in application (①), architecture (②), and mapping description (③). The architecture/platform consists of pre-existing IP components like processors, buses, hardware accelerators and memories, while the application describes how these resources are used to implement certain system functionality. For the specification of different domain-specific application workload scenarios we propose a system input stimuli specification (⑤).

The most important property of the proposed framework is that timing and power characterisation is independent from application specification and development. This separation allows platform providers to offer timing and power characterised virtual platform component models. Together with the estimated SW and custom HW components (①) these can be used by the virtual system generation step (⑥) to build a timing and power aware executable virtual system prototype (⑦).

In our parallel application description model we perform a separation of behaviour (computation) and protocol (communication). Our concurrent building blocks are tasks or processes that contain both a behavioural and a protocol part separately. The behaviour part describes the function or algorithm to be executed, written in C/C++ code. This description is independent from an implementation in either HW or SW. Behaviours can be composed of functions and describe a pure sequential execution order. This enables reuse of existing software descriptions. Moreover, most commercial high-level synthesis tools work with C/C++ input.

In order to examine and analyse the parallel application description model under a certain workload scenario, it needs to be stimulated appropriately. The system stimuli might originate from user interaction or communication with other components that are part of the system’s environment.

The platform description model is composed independently from the application model. It is a pure structural and non-executable representation of the execution platform consisting of execution resources (like SW processors, DSPs or ASICs), memories, communication resources (like shared buses), and pre-existing IP components. In addition, constraints that have a direct influence on the timing and power consumption can be annotated.

A realistic system consists of components of different type e.g., custom hard- and software as well as IP-components, like communication infrastructure. Each component’s power and timing information is estimated individually, using a existing and sophisticated point tools. Based on this estimation, an augmented C/C++/SystemC description of the component is generated, containing a power and timing model of the component.

In order to allow a fast simulation and estimation of the overall system, a virtual prototype is then generated automatically based on the different, annotated components. State-of-the-art techniques and tools for virtual prototype simulation, like temporal decoupling, can be used to further improve the simulation performance. During system execution the annotated timing and power information is collected. Depending on the workload model, different execution paths leading to different timing and power values are possible. After simulation, the collected information can be illustrated in a power-over-time diagram or can be used for a power-breakdown.

The annotations can be traced at different levels of granularity to allow a user-defined trade-off between simulation speed and accuracy at task level. Moreover, it is possible to define certain regions of interest in the sequential behaviour of a task that can be investigated down to basic block granularity. The simulation and analysis of timing and power traces enables an evaluation of the chosen application mapping, the performance of the architecture and the effects of the synthesis constraints. Different design configurations or iterations with adjusted mapping, platform composition and constraints allow multi-objective design-space exploration including the derivation of dynamic power management policies.

By building upon existing approaches and industrial-strength point-tools, the proposed integrated framework leads towards a common and unified flow for timing and power aware rapid prototyping and can provide a seamless path to design-space exploration.

REFERENCES


Motivation - Rising heterogeneity and complexity of embedded systems leads to certain gaps and defines challenges a leading industry has to face.

✓ handling complexity of execution platforms and applications
✓ respecting the uncertainty of platform selection and application to platform mapping
✓ finding the balance between increasing power consumption, possible performance, and explicit application needs
✓ meeting memory demands both in size and access times

Urgently Needed: Highly efficient and productive design methodology and a holistic framework for iteratively exploring embedded HW/SW applications.

Augmentation of well established ESL tools enabling performance & power aware virtual prototyping from a combined HW/SW perspective.

The COMPLEX Flow & Toolchain