A High-Level Power Model for MPSoC on FPGA

Roberta Piscitelli, *Student Member, IEEE,* and Andy D. Pimentel *Senior Member, IEEE*
Informatics Institute, University of Amsterdam, The Netherlands
Email: {r.piscitelli,a.d.pimentel}@uva.nl

Abstract—This paper presents a framework for high-level power estimation of multiprocessor systems-on-chip (MPSoC) architectures on FPGA. The technique is based on abstract execution profiles, called event signatures. As a result, it is capable of achieving good evaluation performance, thereby making the technique highly useful in the context of early system-level design space exploration. We have integrated the power estimation technique in a system-level MPSoC synthesis framework. Using this framework, we have designed a range of different candidate MPSoC architectures and compared our power estimation results to those from real measurements on a Virtex-6 FPGA board.

Index Terms—High-level power estimation, system-level MPSoC design space exploration, MPSoC on FPGA.

1 Introduction

The complexity of modern embedded systems, which are increasingly based on MultiProcessor-SoC (MPSoC) architectures, has led to the emergence of system-level design. System-level design tries to cope with the design complexity by raising the abstraction level of the design process. Here, a key ingredient is the use of high-level modeling and simulation to capture the behavior of system components and their interactions, and thereby facilitating early system-level design space exploration (DSE). An important element of such system-level DSE is the high-level modeling for architectural power estimation.

The traditional practice for embedded systems evaluation often combines two types of simulators, one for simulating the programmable components executing software and one for the dedicated hardware parts. However, using such hardware/software co-simulation during the early design stages has major drawbacks: (i) it requires too much effort to build them, (ii) they are often too slow for exhaustive explorations, and (iii) they are inflexible in quickly evaluating different hardware/software partitions. To overcome these shortcomings, a number of high-level modeling and simulation environments have been proposed in recent years. An example is our Sesame system-level modeling and simulation environment [12], which aims at efficient design space exploration of embedded multimedia system architectures. Until now, the Sesame framework has mainly focused on the system-level performance analysis of multimedia MPSoC architectures. So, it did not include system-level power modeling and estimation capabilities. In [14], we initiated a first step towards this end, however, by introducing the concept of *computational event signatures*, allowing for high-level power modeling of microprocessors (and their local memory hierarchy). This signature-based power modeling operates at a higher level of abstraction than commonly-used instruction-set simulator (ISS) based power models and is capable of achieving good evaluation performance. This is important since ISS-based power estimation generally is not suited for early DSE as it is too slow for evaluating a large design space: the evaluation of a single design point via ISS-based simulation with a realistic benchmark program may take in the order of seconds to hundreds of seconds. Moreover, unlike many other high-level power estimation techniques, the signature-based power modeling technique still incorporates an explicit micro-architecture model of a processor, and thus is able to perform micro-architectural DSE as well.

In this paper, we extend the aforementioned signature-based power modeling work, and we present a full system-level MPSoC power estimation framework based on the Sesame framework, in which the power consumption of all the system components is modeled using signature-based models. The MPSoC power model has been incorporated into Daedalus, which is a system-level design flow for the design of MPSoC based embedded multimedia systems [15]. Daedalus offers a fully integrated tool-flow in which system-level synthesis and FPGA-based system prototyping of MPSoCs are highly automated. This allows us to quickly validate our high-level power models against real MPSoC implementations on FPGA.

The next section briefly introduces the Sesame framework and describes the concept of using *event signatures* for power modeling of architectures. Section 3 gives an overview of our MPSoC power modeling framework. Section 4 presents a number of experiments in which we compare the results from our models against real measurements of real MPSoC implementations on an FPGA. In Section 5, we describe related work, after which Section 6 concludes the paper.

2 Event Signatures

Our system-level power modeling framework [13] is based on the Sesame MPSoC simulation framework [12]. Sesame recognizes separate application and architecture models within a system simulation. An application model describes the functional behavior of a (set of) concurrent application(s). An architecture model defines architecture resources and captures their performance constraints. Subsequently, using a mapping model, an application model is explicitly mapped onto an architecture model. Hereafter, the application and architecture models are co-simulated via trace-driven simulation. That is, by executing an application model it generates traces of application events, which are an abstract representation of the workload (both in terms of computation and communication) that is imposed on the

Posted to the IEEE & CSDL on 10/14/2011
DOI 10.1109/L-CA.2011.24 1556-6056/11/$26.00 © 2011 Published by the IEEE Computer Society

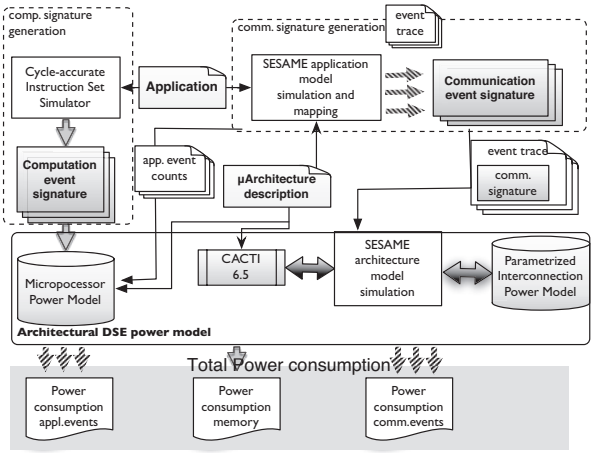

Fig. 1. System-level Power estimation framework

underlying MPSoC architecture model. Subsequently, the architecture model simulates the performance consequences of these computational and communication events generated by the application model.

To extend Sesame with system-level power modeling, we propose a high-level power estimation method based on so-called event signatures. Signature-based power estimation provides an abstraction of processor activity and communication in comparison to e.g. traditional ISS-based power models, while still incorporating an explicit micro-architecture model and thus being able to perform micro-architectural DSE. An event signature is an abstract execution profile of an application event that describes the computational complexity of an application event (in the case of computational events) or provides information about the data that is communicated (in the case of communication events). Hence, it can be considered as meta-data about an application event.

2.1 Computational events signatures

A computational signature describes the complexity of computational events in a (micro-)architecture independent fashion using an Abstract Instruction Set (AIS) [14].

To construct the signatures, the real machine instructions that embody an application event (derived from an ISS) are first mapped onto the AIS, after which a compact execution profile is made. This means that the resulting signature is a vector containing the instruction counts of the different AIS instructions. The event signatures act as input to our parameterized microprocessor power model, which will be described in more detail in the next section.

2.2 Communication event signatures

In Sesame, the application tasks generate *read* and *write* communication events as a side effect of reading data from or writing data to ports. Hence, communication events are automatically generated. For the sake of power estimation, the communication events are also extended with a signature. A communication signature describes the complexity of transmitting data through a communication channel (e.g., FIFO, Memory Bus, PLB Bus) based on the dimension of the transmitted data and the statistical distribution of the contents of the data itself. For the latter, we use the *average Hamming distance* of the coarse-grained data communications. More specifically, we calculate the average Hamming distance of the data words within the data chunk communicated by a *read* or *write* event (which could be, e.g., a pixel block, or even an entire image frame), after which the result is again averaged with the Hamming distance of the previous data transaction on the same communication channel. In this way, we can get information of the usage of the channel and the switching factor, which is related to the data distribution.

2.3 Signature-based, system-level power estimation

In Figure 1, the entire signature-based power modeling framework is illustrated. More specifically, an application task model is mapped onto a given architecture model, and simulated with Sesame [12]; during this stage, each task generates its own trace of application events. The communication signature generation is mapping dependent: communication patterns change with different mappings. For every *read*/*write* event, the *average Hamming distance*, as explained in the previous subsection, is computed. This information is then integrated in the trace events, forming the communication signature. On the other side, the application processes for which a power estimation needs to be performed, are simulated using the ISS, constructing the computational event signatures. The Sesame architecture model simulates the performance and power consequences of the computation and communication events generated by the application model. To this end, each architecture model component is parameterized with an event table containing the latencies of the application events it can execute [12]. Moreover, each architecture model component now also has an underlying signature-based power model. These models are activity-based for which the activity counts are derived from the different application events in the event traces as well as the signature information of the separate events. The total power consumption is then obtained by simply adding the average power contributions of microprocessor(s), memories and interconnect(s).

3 Power Models

We propose a high-level power estimation method based on the previously discussed event signatures that allows for flexible power estimation in the scope of system-level DSE. As will be explained in the subsequent subsections, signature-based power estimation provides an abstraction of processor (and communication) activity in comparison to e.g. traditional ISS-based power models, while still incorporating an explicit micro-architecture model and thus being able to perform micro-architectural DSE. The power models are based on FPGA technology, since we have incorporated these models in our system-level MPSoC synthesis framework Daedalus [15], which targets FPGA-based (prototype) implementations. The MPSoC power model is formed by three main building blocks, modeling the microprocessors, the memory hierarchy and the interconnections respectively. The model is based on the activity counts that can be derived from the application events and their signatures as described before, and on the power characteristics of the components themselves, measured in terms of LUTs used. In particular, we estimate through synthesis on FPGA the maximum number of LUTs used for each component. The

resulting model is therefore a compositional power model, consisting of the various components (for which the models are described below) used in the MPSoC under study. In the remainder of this paper, we will focus on homogeneous systems, but the used techniques do allow the modeling and simulation of heterogeneous systems as well.

3.1 Interconnection Power model

In this section, we derive architectural-level parameterized, activity based power models for major network building blocks within our targeted MPSoCs. These include FIFO buffers, crossbar switches, buses and arbiters. The currently modeled building blocks – network components as well as processor and memory components – are all part of the IP library of our Daedalus synthesis framework [15], which allows the construction of a large variety of MPSoC systems. Consequently, all our modeled MPSoCs can actually be rapidly synthesized to and prototyped on FPGA, allowing us to easily validate our power models. Our network power models are composed of models for the aforementioned network building blocks, for which each of them we have derived parameterized power equations. These equations are all based on the common power equation for CMOS circuits:

$$P_{interconnect} = V_{dd}^2 fC\alpha \quad (1)$$

where f is the clock frequency, V_{dd} the operating voltage, C the capacitance of the component and α is the average switching activity of the component respectively. The capacitance values for our component models are obtained through an estimation of the number of LUTs used for the component in question as well as the capacitance of a LUT itself. Here, we estimate the number of LUTs needed for every component through synthesis, after which the capacitance is obtained using the X-Power tool from Xilinx[17]. The activity rate α is primarily based on the *read* and *write* events from the application event traces that involve the component in question. For example, for an arbiter component of a bus, the total time of read and write transactions to the bus (i.e., the number of *read* and *write* events that involve the bus) as a fraction of the total execution time is taken as the access rate (i.e., activity rate). For communication channels like busses, not only the number of *read* and *write* events play a role to determine the activity factor, but also the data that is actually communicated. To this end, we consider the *Hamming Distance distribution* between the data transactions, as explained in the previous section on communication signatures. Thus, every communication trace event is carrying the statistical activity-based information of the channel from/to which the data is read/written. Consequently, for any activity (read/write of data) in the channel, the dynamic power of the interconnection is calculated according to technology parameters and the statistical distribution of the data transmitted.

In our models, leakage power is calculated according to the estimated look-up tables needed to build a particular interconnection.

3.2 Memory and Microprocessor Power model

For on-chip memory (level 1 and 2 caches, register file, etc.) and main memory, we use the analytical energy model developed in CACTI 6.5 [10] to determine the power consumption of read and write accesses to these structures. These power estimates include leakage power. The access rates for the processor-related memories, such as caches and register file, are derived from the computational signatures. For the main memory and communication buffers, we calculate the activity rate by means of the access rate and the switching probability of the signals. For every read/write event to the memory, the average Hamming distance contained in the communication event signature is extracted and the signal rate is calculated.

The microprocessor model that underlies our power model is based on [14]. It assumes a dynamic pipelined machine, consisting of one arithmetic logical unit, one floating point unit, a multiplier and two levels of caches. However, this model can easily be extended to other processor models, by simply introducing new units. The power consumption of a computational application event is calculated by accumulating the power consumption of each of the components that constitute the microprocessor power model. The microprocessor power model uses a micro-architecture description file in which the mapping of AIS instructions to usage counts of microprocessor components is described. Using this description, power consumption estimates for each computational application event are produced by accumulating the power consumption of the processor components used by the application event.

4 Validation

As mentioned before, we have integrated our power model into the Daedalus system-level design flow for the design of MPSoC based embedded multimedia systems [15]. This allows for direct validation and calibration of our power model. By deploying Daedalus, we have designed several different candidate MPSoC configurations and compared our power estimates for these architectures with the real measurements. The studied MPSoCs contain different numbers of Microblaze processors that are interconnected using a crossbar network. The results of the validation experiments are shown in Figure 2. In these experiments, we mapped three different parallel multimedia applications onto the target MPSoCs: a Motion-JPEG encoder (Mjpeg), a Periodogram, which is an estimate of the spectral density of a signal, and a Sobel filter for edge detection in images. In addition, for each of the applications, we also investigated two different task mappings onto the target architectures. Here, we selected one "good" mapping, in terms of task communication, as well as a "poor" one for each application. That is, in the "good" mapping we minimize task communications, while in the "poor" one we maximize task communications. The experiments in Figure 2 apply the following notation: $app_{name}\_n_{proc}\_mapping_{type}$, where app_{name} is the application considered, n_{proc} indicates the number of processors used in the architecture, and $mapping_{type}$ refers to the type of mapping used. The power values in Figure 2 are scaled by a factor of 2W for the sake of improved visibility.

The results in Figure 2 show that our power model performs decently in terms of absolute accuracy. We observed an average error of our power estimations of around 7%, with a standard deviation of 5%. More important in the context of early design space exploration, however, is the fact that our power model appears to be very capable of estimating the right power consumption trends for the

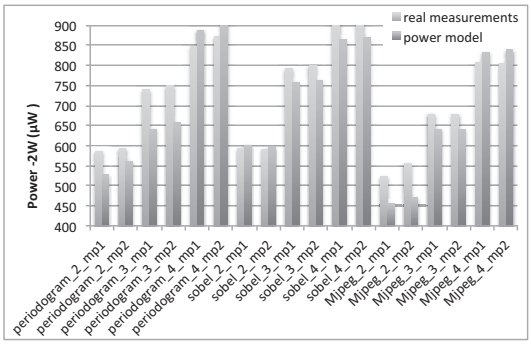

Fig. 2. Mjpeg , Sobel and Periodogram applications with different mappings.

various MPSoC configurations, applications and mappings. Our estimates result in a ranking of the power values that is correct for every application we considered, therefore showing a high fidelity. This high-fidelity quality-ranking of candidate architectures thus allows for a correct candidate architecture generation and selection during the process of design space exploration.

Since every design point evaluation takes only 0.16 seconds on average, the presented power model offers remarkable potentials for quickly experimenting with different MPSoC architectures and exploring system-level design options during the very early stages of design.

5 Related Work

There exists a fairly large body of related work on system-level power modeling of MPSoCs, like e.g. [4], [2], [9], [16], [7]. Moreover, there also exist a considerable number of research efforts that only focus on the power modeling of the on-chip network of MPSoCs. Examples are [11], [5], [6], [8]. Many of these approaches calibrate the high-level models with parameters extracted from RTL implementations, using low-level simulators for the architectural components. To the best of our knowledge, none of the existing efforts have incorporated the power models in a (highly automated) system-level MPSoC synthesis framework, allowing for accurate and flexible validation of the models. Instead, most existing works either use simulation-based validation (e.g. [4], [5], [6], [3], [11]), or validation by means of measurements on fixed target platforms (e.g. [16], [7]). Consequently, in general, related system-level MPSoC modeling efforts do also not target FPGA technology in their system-level power models.

6 Conclusion

We presented a framework for high-level power estimation of multiprocessor systems-on-chip (MPSoC) architectures on FPGA. The technique is based on abstract execution profiles called "event signatures", and it operates at a higher level of abstraction than, e.g., commonly-used instruction-set simulator (ISS) based power estimation methods and should thus be capable of achieving good evaluation performance. The signature-based power modeling technique has been integrated in our Daedalus system-level MPSoC synthesis framework, which allows a direct validation and calibration of the power model. We compared the results from our signature-based power modeling to those from real measurements on a Virtex 6 FPGA board; in particular, we use an I2C controller in the PMBus controller chip [1] included in the FPGA board, in order to measure the power in the FPGA itself [13].

These validation results indicate that our high-level power model achieves fairly accurate power estimates. As future work, we plan to perform more extensive validation experiments (e.g., using different interconnects and memory hierarchies) as well as to deploy the power model in real system-level DSE experiments.

Acknowledgment

This work has been partially supported by the MADNESS STREP-FP7 European Project.

References

[1] "http://pmbus.org/specs.html."
[2] S. D. J.-L. B. Atitallah, R. Niar, "MPSoC power estimation framework at transaction level modeling," *ICM 2007*.
[3] N. Eisley, V. Soteriou, and L. Peh, "High-level power analysis for multi-core chips," in *CASES '06: Proc. of the 2006 int. conference on Compilers, architecture and synthesis for embedded systems*, USA, 2006, pp. 389–400.
[4] D. S. G. Beltrame and C. Silvano, "Multi-accuracy power and performance transaction-level modeling," in *DATE '08: Proc. of the conference on Design, automation and test in Europe*.
[5] J. Hu and R. Marculescu, "Energy-aware mapping for tile-based noc architectures under performance constraints," in *ASP-DAC '03: Proc. of the 2003 Asia and South Pacific Design Automation Conference*, USA, 2003, pp. 233–239.
[6] S. Koohi, M. Mirza-Aghatabar, S. Hessabi, and M. Pedram, "High-level modeling approach for analyzing the effects of traffic models on power and throughput in mesh-based nocs," in *VLSID '08: Proc. of the 21st Int. Conference on VLSI Design*.
[7] I. Lee, H. Kim, P. Yang, S. Yoo, E. Chung, K. Choi, J. Kong, and S. Eo, "Powervip: Soc power estimation framework at transaction level," in *Proc. of the 2006 ASP-DAC '06*. IEEE Press, 2006.
[8] M. Loghi, L. Benini, and M. Poncino, "Power macromodeling of MPSoC message passing primitives," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, p. 31, 2007.
[9] M. Monchiero, G. Palermo, C. Silvano, and O. Villa, "A modular approach to model heterogeneous MPSoC at Cycle Level," in *DSD '08: Proc. of the 2008 11th EUROMICRO Conf. on Digital System Design Architectures, Methods and Tools*, 2008.
[10] N. Muralimanohar, R. Balasubramonian, and N. Jouppi, "Optimizing nuca organizations and wiring alternatives for large caches with cacti 6.0," in *MICRO 40: Proc. of the 40th Annual IEEE/ACM Int. Symposium on Microarchitecture*, 2007.
[11] L. Ost, G. Guindani, L. Indrusiak, S. Maatta, and F. Moraes, "Using abstract power estimation models for design space exploration in NoC-based MPSoC," *IEEE Design and Test of Computers*, vol. 99, no. PrePrints, 2010.
[12] A. D. Pimentel, C. Erbas, and S. Polstra, "A systematic approach to exploring embedded system architectures at multiple abstraction levels," *IEEE Trans. Comput.*, vol. 55, no. 2, 2006.
[13] R. Piscitelli and A. Pimentel, "A high-level power model for mpsoc on fpga," in *Proc. of the 18th Reconfigurable Architectures Workshop (RAW '11)*, 2011.
[14] P. Stralen and A. D. Pimentel, "A high-level microprocessor power modeling technique based on event signatures," in *Proc. of the IEEE/ACM/IFIP Workshop on Embedded Systems for Real-Time Multimedia (ESTIMedia '07)*, 2007.
[15] M. Thompson, H. Nikolov *et al.*, "A framework for rapid system-level exploration, synthesis, and programming of multimedia MPSoCs," in *Proc. of the IEEE/ACM int. conference on Hardware/software codesign and system synthesis*, 2007.
[16] A. Varma *et al.*, "Accurate and fast system-level power modeling: An xscale-based case study," *ACM Trans. Embed. Comput. Syst.*, vol. 6, no. 4, p. 26, 2007.
[17] Xilinx, "http://www.xilinx.com/products/design_tools/logic_design/verification/xpower.htm."