

An Approach for High-Level Thermal Modeling using Native Simulation

D.Calvo, E.Villar

*Microelectronics Engineering Group
University of Cantabria
{daniel, evillar} @teisa.unican.es*

A.Acquaviva, E.Macii

*Department of Automation and Information
Politecnico di Torino
{andrea.acquaviva, enirico.macii} @polito.it*

1. Introduction

In low-power embedded systems, choosing the optimum design alternative is essential in order to fulfil consumption requirements. Leakage energy represents up to 25% of the total energy consumption in nano Complementary Metal Oxide Semiconductor (CMOS) technologies. Among other parameters, high temperatures on the chip increase leakage power. Different floorplans of the same System on Chip (SoC) can lead to designs with 3X difference in leakage power [1]. Thus, taking into account temperature estimations in early Design Space Exploration (DSE) is important to achieve a reliable and optimized design, allowing the inclusion of the appropriate countermeasures at the architectural and platform level.

Nevertheless, low power consumption is not the only reason for developing thermal-aware systems. All the components have an operating temperature range and working above it may drastically reduce the lifespan indicated in the corresponding datasheets. The reduction in lifespan of the different components will also cause the loss of system reliability, which is crucial in most applications of embedded systems. An early detection of potential thermal problems will reduce the possibilities of having to redesign the complete system in the final stages of the design flow.

In order to take temperature into account, advanced thermal modeling is required. The goal of the thermal model is to simulate the temperature for given architectures. Thermal models are fed using power consumption figures. In order to obtain the power estimations, HW/SW simulators are needed since software execution impact must also be considered. Moreover, design space explorers also require simulation engines to perform the exploration, since other performance parameters must also be considered. In most cases, these tools are the bottleneck for the platform exploration.

Native simulation methodology allows modeling and fast simulation of the execution of software applications on the hardware platform in close interaction with all the platform components. Fast system-level simulation enables to study, in reasonable time, thermal effects that are typically characterized by timescales on the order of milliseconds to seconds of real execution. This solution requires the use of advanced models of the main components that make up a SoC, being much faster than the ISS (Instruction-set Simulator) approach, which is the methodology most widely used in HW/SW co-simulation.

Although some work has been done focusing on estimating MPSoCS temperatures [2], it is based on ISS models. Thermal modeling has not been taken into account yet in native simulation approaches, which is the most appropriate alternative for DSE due to the fast simulations and the accurate results [3]. In this work, we are developing an approach to perform thermal studies using native simulation.

2. MPSoC Thermal Analyze

To quickly estimate the temperature of the main components composing a MPSoC, the simulation environment that figure 1 presents is being developed. To simulate real software behavior within a SystemC [4] environment, it is necessary to obtain an executable model including the execution time estimations. The proposed approach uses a two-step model. First, the stand-alone execution time of SW tasks is obtained. In parallel to time estimations, power consumption required by the processor to execute the application software is estimated. These values are used in the second step, where they are back-annotated in the application code by inserting some extra code. The annotated source code is simulated with other component models (processor, instructions cache, memory), thus obtaining a HW-SW co-simulation

environment which provides early performance and power estimations.[5]

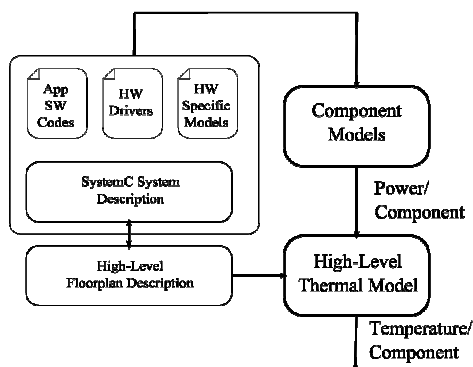


Figure 1. Simulation Environment to estimate MPSoC temperature

To estimate temperatures, the SystemC main file, which depicts the system that will be simulated, must contain additional information. For each CPU, instruction cache or memory instantiated, the position that each element has in the floorplan and its size must be defined. This information is indicated by means of a high-level floorplan [2]. The thermal model is fed using the power estimations that the component models enable to obtain, estimating the temperature of the different components that make up the SoC in a dynamic way.

3. Preliminary Results and Future Work

The current capabilities of the resulting framework are shown with an example: a H264 coder. The system consists of an ARM9 core with 16KB instructions cache and a 512MB RAM memory.

Two floorplans are going to be applied to demonstrate the influence of component positions in the heat flow:

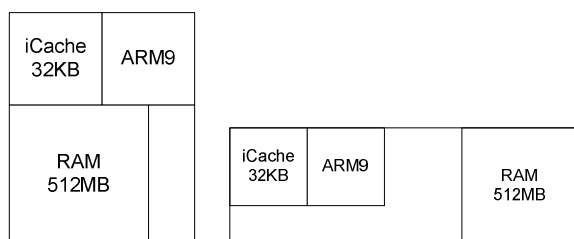


Figure 2. Floorplans

A package to air conductivity of 40K/W has been used. In figure 3, the resulting temperature estimations for the processor are presented.

The accuracy and speed-up of this framework is being

currently evaluated with respect to a more precise system based on an ISS simulation approach.

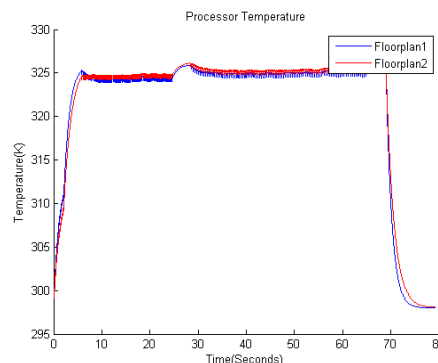


Figure 3. Processor Temperature Estimation

4. Conclusions

The resulting framework of this work will enable fast estimations of the heat flow in MPSoC systems to be obtained, being an appropriate approach for DSE exploration with thermal analysis. As has been shown in the preliminary results, chip temperatures are influenced by the position of the components in the floorplan. Using our approach, it will be possible to rule out early those design alternatives that could result in implementations with potential hot spots problems. Hot spots causes the increase of dynamic power consumption and the loss of component reliability. This preliminary work shows that software simulation is promising technology for high-level thermal modeling.

5. References

- [1] A. Gupta, N.Dutt, F.Kurdahi, K.Khoury, M.Abadir, "Floorplan driven leakage power aware IP-based SoC design space exploration", CODES+ISSS'06, October 22-25, 2006.
- [2] G. Paci, P.Marchal, F. Poletti, L. Benini, "Exploring temperature-aware design in low-power MPSoCs", Proceedings -Design, Automation and Test in Europe, DATE 1, art. no. 1657006, 2006.
- [3] M.Becker, T.Xie, W.Mueller, G. Di Guglielmo, G. Pravadelli and F.Fummi, "RTOS-Aware Refinement for TLM2.0-Based HW/SW Designs", in DATE 2010.
- [4] Open SytemC Initiative (OSCI) Homepage: <http://www.systemc.org>
- [5] H. Posadas, D. Quijano, J. Castillo, V. Fernández, E. Villar and M. Martínez, "SystemC Platform Modeling for Behavioral Simulation and Performance Estimation of Embedded Systems", in L. Gomes and J. M. Fernandes (Eds.): "Behavioral Modeling for Embedded Systems and Technologies: Applications for Design and Implementation", IGI Global, 2009.