

Early Design Stage Thermal Evaluation and Mitigation: the Locomotiv Architectural Case

Tanguy Sassolas*, Chiara Sandionigi*, Alexandre Guerre*, Alexandre Aminot*,
Pascal Vivet†, Hela Boussetta‡, Luca Ferro ‡, Nicolas Peltier ‡

* CEA, LIST, 91191 Gif-sur-Yvette CEDEX, FRANCE - Email: surname.name@cea.fr

† CEA, LETI, Minatec Campus, Grenoble, FRANCE - Email: surname.name@cea.fr

‡ DOCEA Power, 166 rue du Rocher de Lorzier 38430 Moirans, FRANCE - Email surname.name@doceapower.com

Abstract—To offer more computing power to modern SoCs, transistors keep scaling in new technology nodes. Consequently, the power density is increasing, leading to higher thermal risks. Thermal issues need to be addressed as early as possible in the design flow, when the optimization opportunities are the highest. For early design stages, architects rely on virtual prototypes to model their designs' behavior with an adapted trade-off between accuracy and simulation speed. Unfortunately, accurate virtual prototypes fail to encompass thermal effects timescale. In this paper, we demonstrate that less accurate high-level architectural models, in conjunction with efficient power and thermal simulation tools, provide an adapted environment to analyze thermal issues and design software thermal mitigation solutions in the case of the Locomotiv MPSoC architecture.

I. INTRODUCTION

Transistor size reduction induced greater power density resulting in higher chip temperature issues. Not considering heat dissipation apparatus, the common way to deal with thermal hotspots consists in evenly distributing temperature across the chip by managing the active portions of the circuit. Thus, the usage of the system will dictate its power and temperature evolution. Consequently, the architecture and software designs have to take into account these constraints as early as possible in the design flow. Another issue brought by technology scaling is that the leakage current is becoming a bigger part of the power consumption. The hard part being that the leakage is highly dependent on the temperature, which can lead to thermal runaway effects. So, to efficiently deal with thermal issue we need an Electronic System Level (ESL) design environment that can take into account the system's functionality, its power and thermal behaviors while modelling their mutual influences. In this paper, we present how we traded functional behavior and power accuracy to offer a fast environment to analyze thermal issues and design software thermal mitigation solutions for the Locomotiv MPSoC architecture. The environment is composed of a Programmer's View Loosely-Timed (PVLTL) model tightly coupled with Aceplorer, a commercial ESL power and thermal analysis and optimization tool, and AceThermalModeller, a compact thermal model generation tool, both developed by DOCEA Power.

II. THE LOCOMOTIV ARCHITECTURE

The Locomotiv architecture [1] is a quad STxP70 architecture implemented in 28 nm technology node jointly designed with STMicroelectronics. All processors access a local shared memory through an Asynchronous Network on Chip (ANoC). The chip also includes a Direct Memory Access (DMA) to efficiently retrieve data from external memory and a Hardware Synchronizer (HWS) used to accelerate the synchronizations between the cores and leverage parallel computing power. The ANoC allows for independent local Dynamic Voltage and Frequency Scaling (DVFS) per core. All cores have various

Dynamic Power Modes (DPM), including fetch disable, clock gating, power gating and DVFS with Vdd-Hopping. Additional probes allow to monitor the ageing and temperature of the chip at runtime to find optimal frequency modes for a given power state. The Locomotiv architecture supports a Hardware Assisted Runtime Software (HARS [2]) which includes various parallel programming API adapted to all application kernel sizes. We now focus on the modelling effort that was conducted to take into account its power, its temperature and its behavior.

1) *Power Modelling*: An Aceplorer power model is composed of one or more power states for each component. For every power state, the user provides an analytical model for both leakage and dynamic consumptions. A particular effort was put into modelling the various power modes of the processing units, including a Vdd-Hopping mode dependent on the frequency state. RTL simulations were used to populate the analytical model. The impact of the temperature on both leakage and dynamic currents was modelled as follows: $exp(\beta_{Leak} * (T - T_{ref}))$ for leakage and $1 + \gamma_{Dyn} * (T - T_{ref})$ for dynamic consumption. The β_{Leak} and γ_{Dyn} coefficients were respectively obtained through an exponential and linear regression from the power characterization data obtained for different temperature corner cases.

2) *Thermal Modelling*: To model the thermal behavior of the system, a physical description of the geometry using rectangular cuboids and detailing every composing material with its thermal properties was made in AceThermalModeler. This description was automatically processed to obtain a Dynamic Compact Thermal Model (DCTM). With such a DCTM the evaluation of the temperature is greatly accelerated while keeping a sufficient level of accuracy for ESL thermal evaluation. The DCTM can be imported in Aceplorer to simulate temperature effects. The Locomotiv chip is packaged into an SBGA304 from Amkor. Its geometrical description was conducted along with JEDEC standardized dimensions. The thermal properties of chemical elements were taken from literature while for compound elements, such as protection glue, they were extrapolated from reseller datasheet. The die floorplan is shown in fig.1.

3) *Functional model*: The last part of our environment is the functional simulation. The thermal phenomena for the locomotiv architecture span over several seconds. Thus, to develop and evaluate thermal mitigation scheme we needed a functional simulator that could represent the execution of tens of seconds in just a few minutes. Programmer's View models (PV) comply with this requirement but cannot take into account execution time that are necessary to evaluate the power consumption. On the opposite, accurately-timed models are too slow to provide an efficient thermal mitigation development platform but offer efficient validation framework [3]. As a result, we designed a programmer's view model

TABLE I. EXECUTION TIME ACCURACY OF THE PVLТ MODEL VS EMULATED ARCHITECTURE

Parallelisation	1 core	2 cores	4 cores
KCycles on Zebu-Server	1810	905	452
KCycles on PVLТ model	1654	1055	533
PVLТ model error	-8 %	+16%	+17%

that was loosely timed (PVLТ). Our simulator, based on an x86 implementation of the HARS runtime [4], compiles into a single executable the Locomotiv application, the runtime software, the runtime Hardware Abstraction Layer (HAL) code and the OSCI SystemC library. All the application and runtime code are executed directly on the x86 host. The x86 cycles are monitored during the execution using the RDTSC instruction. The target STxP70 cycles are extrapolated using a rule-of-thumb based on known processors Instructions Per Cycles (IPC). An emulated Locomotiv architecture using a Zebu-Server from Synopsys was used to validate the relative accuracy of our model. A matrix multiplication code executed on a variable number of STxP70 cores was used for that purpose. The results summarized in table I show that the error is kept below 20%. Using AceTLMConnect, a SystemC activity monitoring library, all power state changes, as well as instruction cycles, are monitored in the PVLТ model. This data is sent to Aceplorer which performs the power and temperature simulation. Then, the temperature of the system for every dissipating components is sent back to the PVLТ model. This close loop co-simulation with Aceplorer is compulsory to allow the development of thermal mitigation and to assess their impact on the execution and temperature. The exchange rate between the two simulators is dictated by power mode change and a period set according to the expected thermal phenomena.

III. EVALUATED MITIGATION SCHEMES AND APPLICATION

This complete environnement was used to develop thermal mitigation schemes for a parallel Advance Driver Assistance System (ADAS) application: pedestrian detection. This application is composed of 2 parts. The first one, called preprocessing is very regular in its execution and represents around 17% of the total execution. The second one is composed of cascading classifier executing on different portions of the image and distances. The execution time of this phase varies according to the number of pedestrians in the scene. This application also has strong real time constraints per frame processing. We implemented 2 mitigation schemes and compared them to a standard execution. The standard scenario only switches off processors that are unoccupied for a given time using the most reactive DPM mode. The first mitigation only considers temperature thresholds at which the system shall be switched off ($90^{\circ}C$) or on ($70^{\circ}C$) to stay secure. We observed that switching off the circuit induced a rapid thermal drop that was met by a corresponding rapid increase when switched back on. So we set the low-level threshold low enough to reduce the temperature not only locally but for a portion of the package. The second method implements a slack reclamation algorithm to adapt processor speed according to expected classification work and is insensitive to variation in image complexity. As shown in table II, the first mitigation achieves best thermal reduction but induces many successive skipped frames which is unacceptable. The second mitigation is slightly less thermally efficient but skips fewer frames and none are successive which can be corrected with a pedestrian

TABLE II. THERMAL MITIGATION & SIMULATION SPEED COMPARISON

Mitigation case	Standard	Thresholds	Slack reclamation
Maximal Temperature ($^{\circ}C$)	94.52	90.71	91.2
Frames skipped (over 52)	1	10	3
Simulation time	21 min	15 min 34 s	20 min 2 s

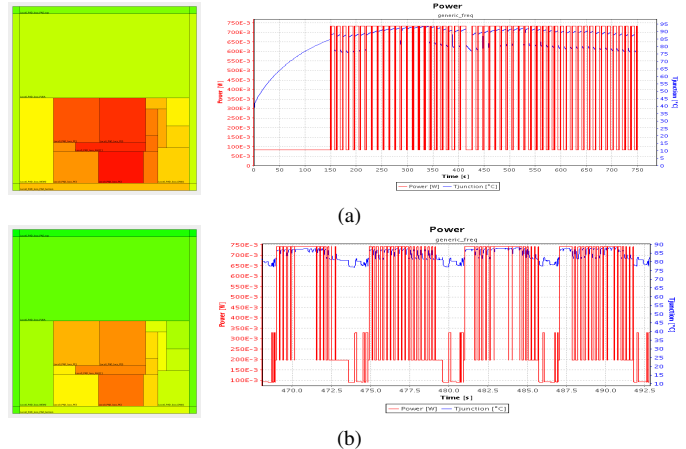


Fig. 1. Power and thermal execution profile for the first processor and worst case die floorplan view comparison between the standard case (1a) and the slack reclamation(1b, focus on 4 frames) temperature mitigation schemes. In red is displayed the power, in blue the temperature.

tracking phase. The execution profiles for the standard case and the slack mitigation are presented in fig 1. In terms of simulation speed, the overall execution of 52 frames took at worst 21 minutes on a core i7-3770 at 3.4GHz which is adapted to thermal software development phases. The lower execution time for the threshold management is explained by the frame skipping which reduces the PVLТ processing.

IV. CONCLUSION

In this paper we proposed to relax the accuracy on both the behavior modelling and the power accuracy to gain in simulation speed and deliver an efficient development environment for thermal mitigation. For a complete pedestrian detection application running on the Locomotiv architecture we were able to compare 2 thermal mitigation schemes applied during 10 simulated minutes for only twice that simulation time. We also showed that trivial thermal management is not sufficient for time-constrained application. So insight on the system functional, power and temperature behaviors is compulsory at the electronic system level. In the future, we plan to extend this co-simulation environment with co-emulation to bring both accuracy and speed for thermal mitigation development later in the flow. We also plan on studying the benefit of our environment for ageing evaluation.

REFERENCES

- [1] E. Beigne et al., "A fine grain variation-aware dynamic Vdd-hopping AVFS architecture on a 32nm GALS MPSoC," in *Proceedings of the European Solid State Circuits Conference*, 2013, pp. 57–60.
- [2] Y. Lhuillier et al., "HARS: A hardware-assisted runtime software for embedded many-core architectures," *ACM Transactions on Embedded Computing Systems*, to be published.
- [3] K. Skadron et al., "Temperature-aware microarchitecture: Modeling and implementation," *ACM Transactions on Architecture and Code Optimization*, vol. 1, no. 1, pp. 94–125, 2004.
- [4] A. Aminot et al., "PACHA: Low Cost Bare Metal Development for Shared Memory Manycore Accelerators," *Procedia Computer Science - ICCS*, vol. 18, pp. 1644–1653, 2013.