Thermal-Aware System-Level Modeling and Management for Multi-Processor Systems-on-Chip

Francesco Zanini†, David Atienza⋆, Luca Benini‡, Giovanni De Micheli†
† Laboratory of Integrated Systems (LSI), EPFL, Switzerland
⋆Embedded Systems Laboratory (ESL), EPFL, Switzerland
‡Micrel Laboratory, DEIS, University of Bologna, Italy

***Abstract*—Multi-Processor Systems-on-Chip (MPSoCs) are penetrating the electronics market as a powerful, yet commercially viable, solution to answer the strong and steadily growing demand for scalable and high performance systems, at limited design complexity. However, it is critical to develop dedicated system-level design methodologies for multi-core architectures that seamlessly address their thermal modeling, analysis and management. In this work, we first formulate the problem of system-level thermal modeling and link it to produce a global thermal management formulation as a discrete-time optimal control problem, which can be solved using finite-horizon model-predictive control (MPC) techniques, while adapting to the actual time-varying unbalanced MPSoC workload requirements. Finally, we compare the system-level MPC-based thermal modeling and management approaches on an industrial 8-core MPSoC design and show their different trade-offs regarding performance while respecting operating temperature bounds.**

I. Introduction

With the advance of technology, the number of functional units and cores integrated on a chip is increasing. Already today several commercial multicore architectures ranging from few cores to several tens of cores are starting to be available, such as Sun's 8-core Niagara [1] and Tilera's 64-core architecture [2]. However, power and thermal management are critical challenges for such multicore systems [4]. In fact, temperature gradients and hot spots not only affect the performance of the system, but also lead to unreliable circuit operation and reduce the lifetime of the chip [3]. In addition, abrupt power-mode transitions in voltage and frequency scaling waste additional power [6]. Thus, system-level thermal management for multicore architectures is a critical matter to tackle nowadays.

In the last years, thermal management techniques received a lot of attention. Many state-of-the-art thermal control policies operate power management by employing *dynamic voltage and frequency scaling* (DVFS) based techniques [5], [6]. On the one hand, most previous works target power density reductions, which has the indirect effect of reducing overall temperature. However, these techniques do not proactively minimize thermal gradients or hot spots, but rather take reactive decisions based on information related to the current thermal profile and frequency setting of the MPSoC to control. On the other hand, most recent approaches start tackling joint processor power optimizations and thermal management by using convex optimization [8], [12], [15]. These works start exploiting a temperature forecast technique based on convex-optimization models, but the adaptation is only done on-line exploiting information related to a limited workload history.

Finally, two recent approaches [16] and [17] describe two methods to achieve thermal prediction without completely relying on the thermal model, thermal sensors and power consumption statistics. However, these previous policies do not completely avoid hot spots, but they simply reduce their frequency. The reason is that the interaction between the prediction method, the thermal behavior of the MPSoC and the frequency assignment of the MPSoC is not addressed as a joint system-level optimization problem. Thus, the actions taken by the policies to avoid hot spots do not explore the problem from a global optimum perspective.

In this work, we explore novel thermal-aware system-level control approaches for multi-core architectures that seamlessly considers thermal modeling, analysis and management. We first define a thermal model of the underlying MPSoC architecture, which is linked to the thermal response of the physical materials and chip geometries, in an initial design-time thermal response analysis of the target MPSoC. Then, we show how these approaches exploit this MPSoC thermal model to solve the system-level run-time thermal management problem as an optimization problem using model predictive control (MPC) [7]. Our results on an 8-core MPSoC design show that different system-level MPC-based thermal control methods have different trade-offs regarding workload demands fulfillment, frequency assignment speeds and thermal control computation overhead.

The remainder of this paper is organized as follows. Section 2 present the formulation of the MPSoC system-level thermal model. Section 3 describes the MPC-based thermal management policies. Section 4 presents the experimental results of the comparisons between the MPC-based thermal management approaches. Finally, Section 5 summarizes the main conclusions of this work.

II. System-Level MPSoC Thermal Modeling

The structure of the considered system-level MPC-based thermal management approaches is reported in the diagram of Figure 1. In these approaches, the thermal management policy regulator monitors the MPSoC, which is partitioned into p islands (or subsystems), each with independent frequency and voltage settings. Then, Vector $\mathbf{f}_\tau \in \Re^\mathbf{p}$ represents the value of the clock frequencies at time τ. The frequency value of input i at time τ is denoted by $(\mathbf{f}_\tau)_\mathbf{i}$. Input i ranges from 1 to p. Thus, the thermal management approach (or policy regulator) sets working frequencies $\mathbf{f}_{\tau+1}$ according to a specific MPC-based control scheme. The frequency setting the regulator does at time τ is performed by taking into account the current frequency setting $\mathbf{f}_\tau$, temperature measurements $\tilde{\mathbf{t}_\tau}$ coming from on-die thermal sensors and a workload requirement coming from the scheduler $\mathbf{w}_\tau$ **epsilon** $\Re^\mathbf{p}$. For each functional unit $i = 1..p$, the workload is defined as the minimum value of the clock frequency that the functional unit should have in order to execute the required tasks within the specified system constraints. The regulator provides a frequency assignment that minimizes the difference between the required and the achieved workload.

A. Frequency Input Model

Using the MPSoC system described in Figure 1, we model synchronous MPSoCs with p clocks that are viewed as the inputs to the system: vector $\mathbf{f}_\tau \in \Re^\mathbf{p}$ represents the value of the clock frequencies at time τ. The frequency value of input i at time τ is $(\mathbf{f}_\tau)_\mathbf{i}$. Clock frequencies are continuous, range from zero to a max frequency

978-1-4244-9472-9/11/$26.00 ©2011 IEEE

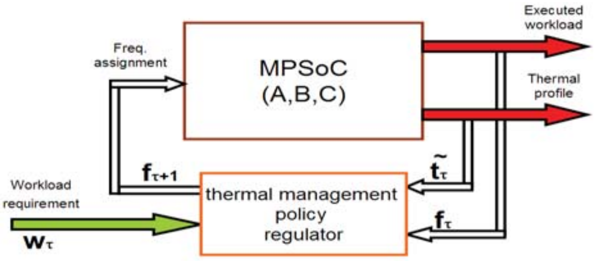

Fig. 1. Diagram of a generic DVFS-based thermal management system

value ($f_{\max}$), and represent our optimization variable. The previous statement is expressed by: $0 \preceq \mathbf{f}_\tau \preceq \mathbf{f}_{\max} \quad \forall\, \tau$, where the symbol $\preceq$ means element-wise comparison, $f_{\max} \cdot \mathbf{1} = \mathbf{f}_{\max}$ and $\mathbf{1}$ is a vector of all ones of size p. Then, the frequency vector represents our optimization variable. The value of its elements is assigned by solving the minimization problem described in the following section, which tries to achieve the desired performance requirements while satisfying the given temperature constraints.

At time τ, the relation between the normalized value of power dissipation $\mathbf{p}_\tau \,\epsilon\, \Re^{\mathbf{P}}$ and the normalized frequency of operation $\mathbf{f}_\tau$ is expressed by: $\mu \mathbf{f}_\tau^\alpha = \mathbf{p}_\tau \quad \forall\, \tau$, where μ is a technology-dependent coefficient. The constant α depends as on the technology as well and usually it takes a value between 1 and 2. If $\alpha = 1$, we have a linear dependence (i.e., frequency scaling) while if $1 < \alpha \leq 2$ we obtain a quadratic or sub-quadratic dependence (i.e., DVFS) [8].

B. Workload Model

The workload requirement is obtained from the higher-level software layers (e.g., operating system). Using the MPSoC system described in Figure 1, for each p islands, the workload is defined as the minimum value of the clock frequency that the functional unit should have to execute the required tasks within the specified system constraints. Thus, the workload requirement at time τ is defined as a vector $\mathbf{w}_\tau \,\epsilon\, \Re^{\mathbf{P}}$, where $(\mathbf{w}_\tau)_\mathbf{i}$ is the workload requirement value for input i at time τ. $(\mathbf{w}_\tau)_\mathbf{i}$ is the frequency that cores associated with input i from time τ to time $\tau+1$ should have to satisfy the desired performance requirement coming from the scheduler. This model is assumed to be continuous and ranges from zero to a max value $\mathbf{f}_{\max}$, the maximum frequency at which the cores process data is:

$$0 \preceq \mathbf{w}_\tau \preceq \mathbf{f}_{\max} \quad \forall\, \tau \tag{1}$$

When $(\mathbf{w}_\tau)_\mathbf{i} > (\mathbf{f}_\tau)_\mathbf{i}$, the workload cannot be processed and so it needs to be stored and re-scheduled in the following clock cycles. The, the way we measure the performance of the system in achieving the requested workload requirements at time τ is given by the vector $\mathbf{u}_\tau \,\epsilon\, \Re^{\mathbf{P}}$, which expresses the undone workload at time τ.

$$\mathbf{u}_\tau = \mathbf{w}_\tau - \mathbf{f}_\tau \tag{2}$$

C. Heat Propagation Model

Our MPSoC system-level thermal model is based on RC-network differentiation [10]. The chip floorplan is divided into thermal cells of cubic shape considering silicon and copper layers, and every functional unit in the floorplan is represented by one or more thermal cells. Temperatures are then computed considering the cells heat conductances and capacitances, and the discretization of the differential equations is solved using a discrete-time, linear, invariant system [14]. The MPSoC thermal model is thus described as follows:

$$\mathbf{x}_{\tau+1} = \tilde{\mathbf{A}}\mathbf{x}_\tau + \tilde{\mathbf{B}}\mathbf{p}_\tau \tag{3}$$

$$\tilde{\mathbf{t}_\tau} = \tilde{\mathbf{C}}\mathbf{x}_\tau \tag{4}$$

Where matrix $\tilde{\mathbf{A}} \,\epsilon\, \Re^{l \times l}$ and matrix $\tilde{\mathbf{B}} \,\epsilon\, \Re^{l \times p}$. The number of states of the new thermal model is l and p is the number of frequency island in the MPSoC. Equation 3 describes the state update for a reduced order model of the MPSoC, as explained in [8], [9], [13]. Then, Matrix $\tilde{\mathbf{C}} \,\epsilon\, \Re^{s \times l}$ in Equation 4 relates the value of the states to temperature measurements in s specific locations inside the MPSoC, and how the measurements can be derived from the state vector $\mathbf{x}$.

III. System-level MPC-based thermal control

All the considered control policies ensure that the maximum MPSoC temperature never exceeds a predefined threshold, while they avoid abrupt frequency and temperature variations both over time and space. Finally, they minimize the undone requested work from the scheduler. Thus, the control problem can be formalized as:

$$\begin{aligned}
J = \quad & \sum_{\tau=1}^{h} \left(\|\mathbf{Q}\mathbf{x}_\tau\|_\mathbf{g} + \|\mathbf{R}\mathbf{p}_\tau\|_\mathbf{j} + \|\mathbf{T}\mathbf{u}_\tau\|_\mathbf{b} \right) && (5)\\
min \;\; J \quad & && (6)\\
subject\ to: \quad & 0 \preceq \mathbf{f}_\tau \preceq \mathbf{f}_{\max} \quad \forall\, \tau && (7)\\
& \mathbf{x}_{\tau+1} = \tilde{\mathbf{A}}\mathbf{x}_\tau + \tilde{\mathbf{B}}\mathbf{p}_\tau \quad \forall\, \tau && (8)\\
& \tilde{\mathbf{C}}\mathbf{x}_{\tau+1} \preceq \mathbf{t}_{\max} \quad \forall\, \tau && (9)\\
& \mathbf{u}_\tau \succeq \mathbf{0} \quad \forall\, \tau && (10)\\
& \mathbf{u}_\tau = \mathbf{w}_\tau - \mathbf{f}_\tau \quad \forall\, \tau && (11)\\
& \mathbf{p}_\tau \succeq \mu \mathbf{f}_\tau^\alpha \quad \forall\, \tau && (12)
\end{aligned}$$

Function J has three sums, where the summation index τ ranges from 1 to h future steps. The system tries to minimize the cost function J and computes the frequency assignment for these steps. The first term $\|\mathbf{Q}\mathbf{x}_\tau\|_\mathbf{g}$ is the g norm of the state vector x weighted by matrix $\mathbf{Q}$, which relates temporal thermal control (hotspot minimization) and spatial thermal control (thermal balancing). The second term $\|\mathbf{R}\mathbf{p}_\tau\|_\mathbf{j}$ is the j norm of the input power vector p weighted by matrix $\mathbf{R}$. The third term $\|\mathbf{T}\mathbf{u}_\tau\|_\mathbf{b}$ is the b norm of the amount of predicted required workload that has not been executed. The weight matrix $\mathbf{T}$ quantifies the importance of workload execution, as required from the scheduler, in the optimization.

Then, regarding the constraints, inequality 7 defines the range of working frequencies that can be used (either a continuous range of frequency settings or a discrete one). Equation 8 defines the evolution of the system according to the present state and inputs. Equation 9 states that temperature constraints should be respected at all times and in all specified locations. Since the system cannot execute jobs that have not arrived, every entry of $\mathbf{u}_\tau$ has to be greater than or equal to 0 as stated by Equation 10. The undone work at time τ, u_τ, is defined by Equation 11. Finally, equation 12 defines the relation between the power vector $\mathbf{p}$ and the working frequencies, where μ is a technology-dependent constant.

This system-level thermal management control problem can then be formulated over an interval of h time steps, which starts at current time τ. For this reason, the approach is said to be predictive. The result of the optimization is an optimal sequence of future control moves (i.e. frequencies and voltages) settings for the cores. Only the first sample of such a sequence is applied and the remaining moves are discarded. At the next time step, a new optimal control problem based on new temperature measurements and required frequencies is solved over a shifted prediction horizon. Hence, this *receding-horizon* [7] mechanism transforms an open-loop design method into a feedback one. Next, we summarize the considered MPC-based thermal management methods.

A. Linear Quadratic Regulator

The *linear quadratic regulator* (LQR) was proposed in [13]. If the maximum MPSoC temperature (Tmax) is less than a certain threshold, the overall control system is a linear feedback system, where MPSoC frequencies are calculated simply by subtracting from the workload requirement $\mathbf{w}_\tau$ the product of the state vector $\mathbf{x}_\tau$ and the controller matrix gain $\mathbf{K}$. The state vector $\mathbf{x}$ is related to the thermal profile by Equation 4. Then, an emergency saturation system ensures that the regulated frequencies go to the minimum value when the maximum MPSoC temperature is higher than the threshold Tmax.

By looking at the general model of Equations 5 - 12, the problem formulation of this policy considers that the horizon is infinite and the reference (the requested workload $\mathbf{w}_\tau$) is assumed to be constant over all this period. Matrix $\mathbf{T}$ is a null matrix and the norm $g = j = 2$. Thus, the objective function is a quadratic form. Because this method is linear, it is not possible initially to link temperature and undone work constraints. However, a bias signal $\mathbf{w_t}$ is added to the control loop to force the system to execute the requested workload. Finally, an emergency saturation mechanism avoids that the maximum MPSoC temperature is higher than the threshold Tmax by clock gating the MPSoC beyond the emergency temperature [13].

B. Explicit/Implicit MPC

This policy, presented for the first time in [9], considers the thermal control problem in MPSoC as an MPC optimal control approach aimed at maximizing a performance metric for a linear dynamic system under input/output constraints. According to the general model of Equations 5-12, this problem formulation considers that the horizon is finite and equal to h and the reference (the requested workload $\mathbf{w}_\tau$) is assumed to be constant over all this period. The state and the power cost function matrices $\mathbf{Q}$ and $\mathbf{R}$ are set to be null. Matrix $\mathbf{T}$ is the identity matrix and the norm $b = 2$. Thus, the objective function is a quadratic form, and the constraints (Equations 7-12) are considered in the problem formulation.

Then, the proposed control strategy can be implemented in two different ways [7]. The first one is called implicit and requires to solve on-line the minimization problem every time the policy is applied. Thus, a significant amount of hardware resources are needed, since the result must be computed in a time frame shorter than the thermal time constants of the MPSoC. An alternative approach is that the MPC problem is solved off-line in a way that makes explicit the dependence of the solution of the frequency assignment problem $\mathbf{f}_\tau^\alpha$ on input vectors $\mathbf{f}_\tau^\alpha$, $\mathbf{w}_\tau^\alpha$ and $\mathbf{x}_\tau$. Bemporad et al. [7] have shown that the optimal explicit controller is *piecewise affine*. The state space can be divided into a set of regions, bounded by linear inequalities (i.e., a polytope), and in each region a different linear controller can be specified and computed off-line. Then, the controller selection can be efficiently performed on-line by simply checking region boundaries.

Finally, a variation of the explicit MPC controller is the approximate explicit MPC approach proposed in in [14]. This method is similar to the explicit approach presented before, but the solution of the optimization problem is computed off-line in a way that makes explicit the dependence of the solution of the frequency assignment problem $\mathbf{f_{\tau+1}}$ on input parameters $\mathbf{w}_\tau$ and $\mathbf{x}_\tau$. The resulting explicit controller is *piecewise polynomial*. Thus, this method provides a significant reduction in hardware requirements and computational cost at the expense of a very limited loss in accuracy (Section IV).

C. Convex Optimization-Based Policies

In this case, the system-level thermal management problem formulation is solved using an embedded solver to compute on-line the frequency assignment [8], [12]. The major difference compared with previous methods is that the algorithm dynamically adapts to the actual run-time situation of the system, without relying on any exhaustive characterization at design time of the possible workloads of the target system. The result of this optimization is a very smooth control where both thermal and reliability constraints are satisfied while achieving significant performance improvements.

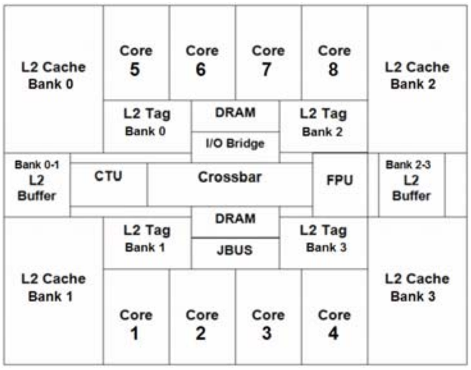

Fig. 2. Floorplan used of the Niagara-1 multicore case study

According to the general model of Equations 5-12, this problem formulation considers a finite horizon and equal to h. The state cost matrix $\mathbf{Q}$ is set to be a null matrix. Matrix $\mathbf{R}$, responsible for the power minimization is an identity matrix and the norm $j = 1$. All the others constraints expressed by Equations 7-12 are considered inside the problem formulation. Then, the major differences with respect to previous problem formulations is that the Matrix $\mathbf{T}$ is time varying, the norm $b = 1$ and the requested workload $\mathbf{w}_\tau$) are time-varying. Thus, the algorithm dynamically predicts the future workload requirements $\mathbf{w}_\tau$ and the reliability of the estimation is embedded in the problem formulation by matrix $\mathbf{T}$.

IV. Experimental Results

The floorplan of the considered MPSoC 8-core Niagara-1 (UltraSparc T1) architecture is shown in Figure 2. It has been modelled using blocks of 3mm side each and the technological parameters have been derived from [1]. We simulated it using a HW/SW emulation infrastructure inspired by [10]. This architecture has a maximum operating frequency of 1.2 GHz, where we consider 10 DVFS settings between 0-1.2GHz in our experiments, and the maximum power consumption of each processor core at the highest frequency is 4 W [1]. We run different web-accessing and multimedia benchmarks [11].

A. Compared system-level thermal management policies

In all our experiments the system-level thermal management policies are applied every $T_{pol} = 10ms$, while the simulation step for the discrete time integration of the RC thermal model (Section II-C) is set to 200μs. The maximum temperature limit is set to $375^\circ K$. The room temperature is set to $300^\circ K$, and we used $\alpha = 2$ [8] to link the DVFS setting and its power consumption.

- Threshold Based DVFS (TB-DVFS): This policy checks if the maximum chip temperature goes above the $370^\circ K$ threshold, and sets the frequency up to 62.5% of the maximum one [5], [8].

- Linear Quadratic Regulator (LQR): This policy is presented in Section II-C and according to our experimental model, the frequency adaptation needs to be performed every T_{pol}.

- Model Predictive Control based policies (MPC): We explore the implicit-explicit optimal MPC approaches presented in [9] and polynomial approximated MPC-based thermal control policies [14] with 100-600 vertices. Our experiments indicate that, comparing

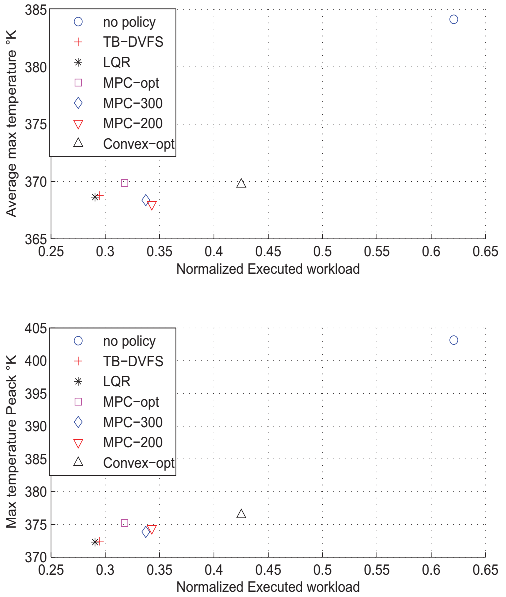

Fig. 3. Executed workload vs. temperature for system-level control policies

with the optimal MPC, approximated controllers reduce the control computation time from $2.7\times$ (600 vertices) to $4.9\times$ (100 vertices). In both cases, the size of the matrices $\mathbf{A}$ and $\mathbf{B}$ is 60x60 and 60x8, respectively, which are reasonable sizes for the on-line computation performed in the OS, using the 10ms target update figure for system-level thermal control. Moreover, even using the approximated controller we can perform a suitable time-scale synchronization between the heat flow propagation model and the MPC approaches, as temperature variation has a much slower time variation constant than the used DVFS and task scheduling tuning knobs.

- Convex Optimization based policy (Convex-opt): The linear predictor has been designed using a 3^{rd} order polynomial equation [12], an observation window of $600ms$ and a prediction length equal to $50ms$ in the future. We assumed to have two frequency inputs controlling the MPSoC: the first one controls cores 1, 4, 5 and 8; the second one sets the frequency value for cores 2, 3, 6 and 7. We suppose that the scheduler tries to perform a workload balancing strategy on the cores and the interval between two consequent applications of the policy ranges from 10ms to 100ms.

B. Trade-offs between executed workload, working temperature, and control computation overhead

In our experiments (Figure 3), we explore the trade-offs of the control policies by plotting the normalized executed workload versus the thermal profile. The top plot analyzes the average maximum chip temperature, while the bottom one shows the maximum MPSoC temperature peak. This figure shows that when no thermal policy is used in our target 8-core MPSoC, even for an average workload of less than 65%, the maximum chip temperature reaches almost 405^oK. On the contrary, most of the system-level MPC-based thermal control policies used in this work are able to avoid this problem (i.e., maximum temperature peak of all policies is less than 377^oK). The best performance in terms of executed workload is provided by convex optimization-based (Convex-opt) policy, followed by the implicit/explicit MPC techniques. All these policy outperforms TB-DVFS in terms of executed workload by a factor of 50%, while having the same average temperature. Then, the approximated MPCs (200-300 vertices) provides almost the same performance as the optimal MPC (MPC-Opt) policy. In fact, the optimum MPC changes the frequencies more often than the approximated versions because it has more DVFS control regions [14]. However, the change in the frequency setting has an overhead in terms of additional power dissipation that is no considered in the presented problem formulation. Thus, this effect leads to an energy loss of approximately 10% compared with the approximated MPC. Moreover, the TB-DVFS and LQR policies show lower performance (50% and 25% less executed workload, respectively) than the convex-optimization approach and the approximated MPCs polices. Then, the LQR policy shows an executed workload that is 3% lower the same compared with the TB-DVFS, which is much simpler, thus outlining that complete MPC-based controllers (implicit and explicit) are better options than LQR policies for system-level thermal control in MPSoCs. Finally, regarding spatial temperature differences, the policy with the smallest thermal variations (both in time and space) is the approximated MPC with 300 vertices and the LQR.

V. Conclusions

MPSoCs have been proposed as a promising solution to provide scalable architectures with limited design complexity, but new thermal-aware design methods must be proposed to guarantee their safe operation. In this paper we introduced the problem of system-level thermal modeling and showed how to formulate it as a discrete-time optimal control problem, which can be solved using finite-horizon model-predictive control (MPC) techniques. Then, we explored different MPC-based techniques on an 8-core Niagara-1 MPSoC and outlined the different trade-offs regarding thermal balancing, workload fulfillment and thermal control overhead with respect to more classical DVFS-based thermal control.

Acknowledgment

The authors thank Dr. Colin N. Jones for his help with the MPC theory. This work has been partially funded by the PRO3D FP7-ICT-248776 project.

References

[1] P. Kongetira et al., *Niagara: A 32-way multithreaded SPARC processor.*, IEEE Micro, 2005.

[2] *Tilera's 64-core architecture*, www.tilera.com/products/processors.php

[3] O. Semenov et al., *Impact of self-heating effect on long-term reliability and performance degradation in CMOS circuits*, IEEE T-D&M, 2006.

[4] S. Borkar, *Design challenges of technology scaling*, IEEE Micro, 1999.

[5] J. Donald et al., *Techniques for multi-core thermal management: Classif. and new exploration*, Proc. ISCA, 2006.

[6] H. Jung et al., *Continuous Frequency Adjustment Technique Based on Dynamic Workload Prediction*, Proc. VLSI Design, 2008.

[7] A. Bemporad et al., *The explicit linear quadratic regulator for constrained systems*, Automatica, 2002.

[8] S.Murali et al., *Temperature Control of High Performance Multicore Platforms Using Convex Optimization*, Proc. DATE, 2008.

[9] F.Zanini et al., *Multicore Thermal Management with Model Predictive Control*, ECCTD 2009.

[10] D.Atienza et al., *HW-SW Emulation Framework for Temperature-Aware Design in MPSoCs*, TODAES, 2007.

[11] Coskun A.K., et al., *Temperature management in multiprocessor SoCs using online learning*, Proc. of DAC 2008.

[12] F.Zanini et al., *Online Convex Optimization-Based Algorithm For Thermal Management of MPSoCs*, GLSVLSI, 2010.

[13] F.Zanini et al., *A Control Theory Approach for Thermal Balancing of MPSoC*, ASPDAC, 2009.

[14] F.Zanini et al., *Multicore thermal management using approximate explicit Model Predictive Control*, ISCAS 2010.

[15] Y. Wang et al., *Temperature-constrained power control for chip multiprocessors with online model estimation*, ISCA 2009.

[16] R.J.Cochran, et al., *Consistent Runtime Thermal Prediction and Control Through Workload Phase Detection*, DAC 2010.

[17] Y.Zhang, et al., *Adaptive and Autonomous Thermal Tracking for High Performance Computing Systems*, DAC 2010.