International Workshop on Innovative Architecture for Future generation Processors and Systems 2007

Exploring Temperature-Aware Design of Memory Architectures in VLIW Systems

José L. Ayala[1], Anya Apavatjrut[2], David Atienza[3,4], Marisa López-Vallejo[1]
[1]Departamento de Ingeniería Electrónica
Universidad Politécnica de Madrid (Spain)
Email: {jayala, marisa}@die.upm.es
[2]Department of Telecommunication Services and Usage
INSA (Lyon, France)
Email: anya.apavatjrut@insa-lyon.fr
[3]Departamento de Arquitectura de Computadores y Automática
Universidad Complutense de Madrid (Spain)
[4]LSI
Ecole Polytechnique Fédérale de Lausanne (Switzerland)
Email: david.atienza@epfl.ch

Abstract

This paper presents a thermal model to analyze the temperature evolution in the shared register files found on VLIW systems. The use of this model allows the analysis of several factors that have an strong impact on the heat transfer: layout topology, placement and memory accesses. Finally, some relevant conclusions are obtained after analyzing the thermal behavior of several multimedia applications.

1 Introduction

As technology scales, higher power consumption coupled with smaller chip area will result in higher power density, which in turn will lead to higher power temperature on the chip [1, 2]. In fact, extrapolating the changes in microprocessor organization and the device miniaturization, one can project future power density to $200W/cm^2$ [3]. This requires extensive efforts on cooling techniques which have shown to be complex and expensive.

While hardware solutions to temperature management problems are very important, software can also play an important role because it determines the circuit components exercised during the execution and the period of time for which they are used. In particular, compilers and source code transformations determine the data and instruction access patterns of applications, what shapes the power density profile. Also, the topology of the hardware modules and the placement of these components determine the temperature behavior.

In this paper we present a complete parameterized thermal characterization of one of the hardware modules that can be found in a VLIW architecture, the shared register file. The experimental approach analyzes the effect of the topology of this device, as well as the placement in the chip layout, in the temperature behavior if the chip when different applications are run.

The contributions of this paper are:

1. Definition of a mathematical model to analyze the temperature behavior of the registers found inside the register file of a VLIW architecture. This model is integrated in a complete simulator of VLIW architectures in order to use the bus activities and register file accesses as input parameters for the model.

2. Analysis of the effect on the temperature behavior of different register file topologies, module placements and register access pattern.

3. Characterization of several multimedia applications to evaluate the common characteristics in terms of temperature behavior.

This paper is composed as follows: Section 2 presents the previous relevant works in this topic, while our proposed methodology and thermal model are briefly explained in Section 3. Also, some optimization policies are presented

1527-1366/07 $25.00 © 2007 IEEE
DOI 10.1109/IWIA.2007.7

in Section 4. Finally, Section 5 covers some preliminary results.

2 Related Work

In recent years there has been an increasing interest to provide a detailed die temperature distribution [4, 5]. In these works, the authors present different detailed full chip thermal models. All these models have detailed temperature distribution across the silicon die and can that be solved efficiently. However, it has been reported recently that the results achieved by these models present inaccuracy problems [6, 7]. All these works provide an analytical method for studying the thermal distribution in the die of high performance processors but they require the complete knowledge of the layout details. Moreover, these models are constrained to the processor layout and cannot be easily extended to different target architectures.

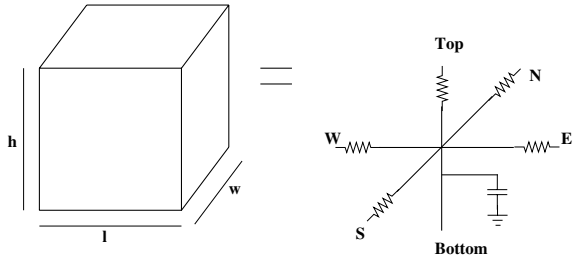
There are other approaches which relay in dynamic measures to characterize the thermal behavior of the chip [8]. These techniques, opposite to ours, require not only the complete knowledge of the target architecture but also the capability to modify it. Some other approaches like [9] or [10] also propose techniques based on electrical measures to develop a power consumption model. However, these works have not dealt with the thermal behavior of the chip. Our previous research work [11, 12] has also targeted the thermal modeling of systems. This paper presents a different approach based on simulation that increases the granularity in the register file of the memory unit and explores several factors with impact on the temperature behavior, like the topology layout or placement.

Our work is based on the thermal model presented at [13,14] and extends its capabilities, modeling the thermal behavior inside the register file when different placements, topologies and running benchmarks are used.

3 Thermal Model

For the development of the thermal model, a well known analogy between the electrical circuits and the thermal sources is exploited. The silicon die and heat spreader is composed in elementary cells in a cubic shape. The temperature for every cell is computed using an RC model. The size of the cell trades-off the simulation speed with the thermal accuracy.

Each cell is associated with a thermal capacitance and five thermal resistances. Four resistances are used to model the horizontal thermal spreading, whereas the fifth is used to model the vertical thermal behavior. The thermal conductivity (horizontal and vertical) and capacitance, respectively, of each elementary cell are computed as follows:



$$G_{hor} = K_{G(Si/Cu)} \times \left(\frac{h \times w}{l}\right)$$

$$G_{ver} = K_{G(Si/Cu)} \times \left(\frac{l \times w}{h}\right)$$

$$C = K_{C(Si/Cu)} \times l \times h \times w$$

where $K_{G(Si/Cu)}$ (thermal conductivity for silicon or copper), $K_{C(Si/Cu)}$ (thermal capacitance per volume unit for silicon or copper), l (cell length), w (cell width), h (cell height).

With this RC characterization, every cell is connected with the cells in the surroundings. The heat dissipation of each block is modeled as a source connected to the current node. A thermal circuit, which is similar to an electrical circuit, is created and can be solved by a node voltage analysis. As a result, the temperature of each block is obtained.

3.1 Register File Modeling

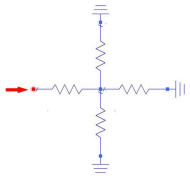
As was mentioned before, one of the goals of this work is to increase the model granularity by focusing the analysis on the temperature behavior of the registers inside the register file. To accomplish such goal, the register file is supposed to be represented as a $N \times M$ matrix and every register belongs to one of the elementary cells. Therefore, the thermal resistance and capacitance (R and C) for every elementary cell and every specific floorplan, have to be calculated.

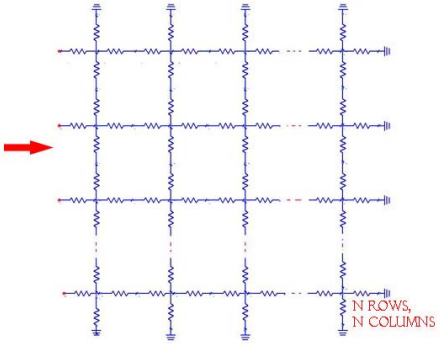
3.1.1 Elementary resistances calculation

Since the total resistance and total capacitance of the device is known in advance, the register file can be decomposed into smaller units. Each unit is associated with its own resistance and capacitance as shown in Figure1.

From Figure 2, the total resistance for a cell is

$$R_{cell} = R + \frac{R}{3} = \frac{4R}{3}$$





Now, from Figure 3, the circuit can be decomposed in a matrix of N rows by M columns of registers, and the resistance of every row can be calculated as follows.

$$\begin{aligned} R_{cell,M} &= \left[(R_{cell,M-1}+R)||\frac{R}{2}\right]+R \\ &= \left[\frac{(R_{cell,M-1}+R)\times\frac{R}{2}}{R_{cell,M-1}+\frac{3R}{2}}\right]+R \end{aligned}$$

Supposing that $R_{cell,M-1} = S_{M-1}R$ and $R_{cell,M} = S_M R$, then

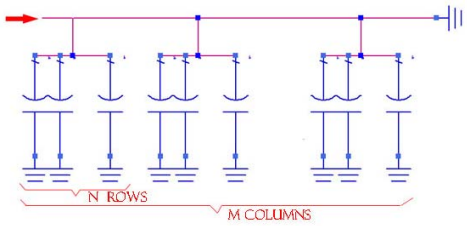
$$S_M = \left\{\left[\frac{S_{M-1}+1}{S_{M-1}+1.5}\right]\times 0.5\right\}+1$$

Considering that each row is parallel with the others, the total resistance for the device can be calculated dividing by the number of rows.

$$R_{tot} = \frac{R_{cell,M}}{N}$$

The resistance of each register can be computed as

$$R = \frac{NR_{tot}}{S_M}$$



3.1.2 Elementary capacitances calculation

The total capacitance of the circuit can be calculated by considering each elementary capacitance to be parallel with the others (see Figure 4).

The total capacitance can be computed for N rows and M columns in parallel as

$$C_{tot} = C \times N \times M$$

The capacitance of every register can be computed as

$$C = \frac{C_{tot}}{N \times M}$$

Once the resistance and capacitance for each elementary cell are known, the size of the elementary cell can be calculated supposing that it is a quadratic cube by the expression

$$size = \frac{R}{K_{G(Si/Cu)}}$$

These last expressions are integrated in the VLSI simulator in order to retrieve the thermal behavior for every register in the register file for different placements and topologies.

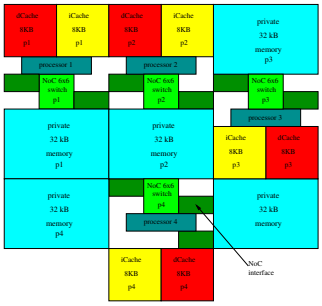
From [13], we employ the same expression to compute the temperature evolution once the technology factors are calculated and the activities are obtained by the simulator.

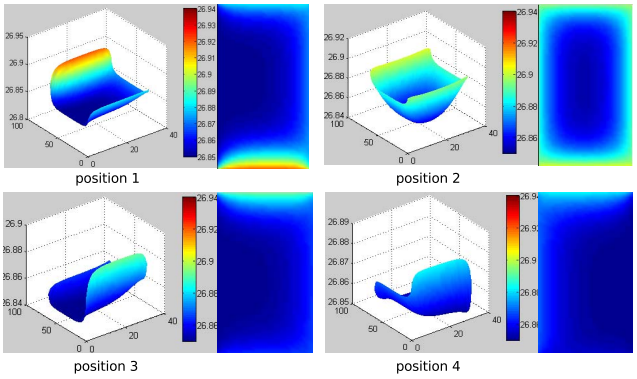
$$\begin{aligned} T_c(n+1)\times 2^{36} &= T_c(n)+((cap\times EC\times 2^{62})\times act)+ \\ &+((A\times 2^{26}-B\times 2^{26}\times(T_c(n))\times \\ &\times(T_n(n)\times 2^{36}-T_c(n)\times 2^{36})))/(2^{26}) \end{aligned}$$

where $T_c(n)$ is the temperature at step n, $T_c(n+1)$ is the temperature at step $n+1$, $T_n(n)$ is the neighbor cell temperature, $cap \times EC$ is the temperature difference due to the activities, act is the activity factor, A is the linear coefficient and B is the quadratic one.

4 Optimization Policies

One of the analysis that has been performed to optimize the thermal behavior of the register file is the selection of





different placement locations for the device. The temperature achieved for every register in the register file is a factor of the number of accesses to the register, the temperature reached by the neighboring registers, and the thermal transfer promoted by the hot functional units close to the register file. All these parameters are considered by the thermal model to characterize the temperature profile.

The baseline architecture devised for the set of experiments resembles a common VLIW system with four processing cores, a shared memory subsystem, a shared register file and a communication network (see Figure 5). The layout of the system is configured in a text file where the placement and size of these modules are coded. One of the configuration files specifies the placement of the layout modules with a letter for every cell ('m' for the memory cell, 'r' for the register file cell, etc.), while the other configuration file provides the size of these cells.

This set of experiments studies the effect of the placement by the selection of four different positions for the register file in the layout of the system. These positions are: close to the processing units (position 1), close to the memory devices (position 2), near the border of the chip (position 3) and surrounded by cooler devices (position 4)

For every one of these placements, the thermal map of the register file is acquired supposing an homogeneous access pattern. Figure 6 shows these thermal maps.

As can be seen, when the register file is surrounded by hot devices as memories (position 2), or close to the processing units (position 1), the temperature of the close registers is increased due to the thermal diffusion and gradients of temperature that can appear between the hotter and cooler devices, destroying the silicon.

On the other hand, when the register file is placed near the border of the chip or it is surrounded by cooler devices (positions 3 and 4), the heat can be transferred to the outside of the system and the temperature is not increased. Moreover, the temperature map of the register file is homogeneous and the thermal gradients are reduced.

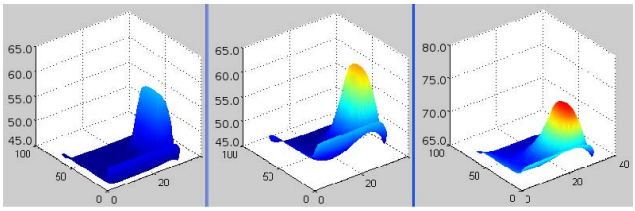
Finally, the last set of experiments analyzes the effect of the access pattern on the temperature of the register file. This analysis will allow to define temperature-aware access policies that reduce the temperature of the device as well as the power consumption [15]. These experiments have been performed for every placement of the layout (positions 1, 2, 3 and 4) and three different access patterns (registers accessed from a bank placed on the right hand side of the register file, accessed registers randomly placed in several spots of the device and registers accessed in a homogeneous manner as a chess board).

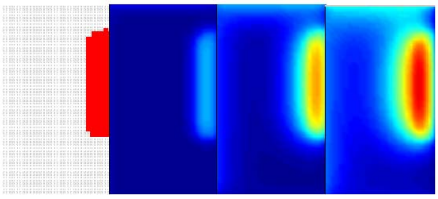
The following graphs show the results for the three different accesses when the register file is placed in position 4.

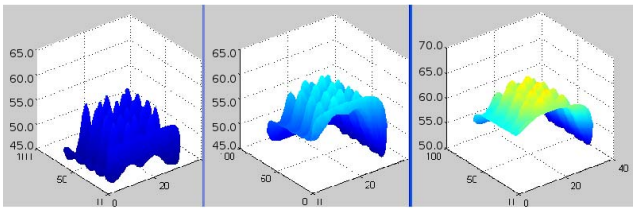
Figure 7 shows the evolution in time of the thermal map for the register file when the registers are accessed from a bank located on the right hand side of the device. As can be seen, the bank where the registers are accessed from is increasingly heated as time advances. At the end, a large hot spot appears in the register file, which can severely damage the device.

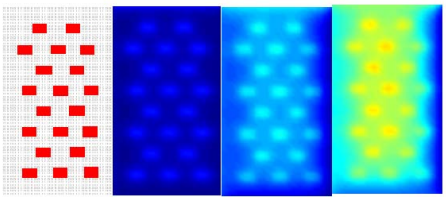
Figure 8 shows the evolution in time of the thermal map for the register file when the registers are randomly accessed from several spots in the device. As can be seen, these spots where the registers are accessed from are increasingly heated as time advances. At the end, several hot spots appear on the register file surface increasing the probability of chip damage. Therefore, an access pattern what homogenizes the thermal map on the silicon surface must be found.

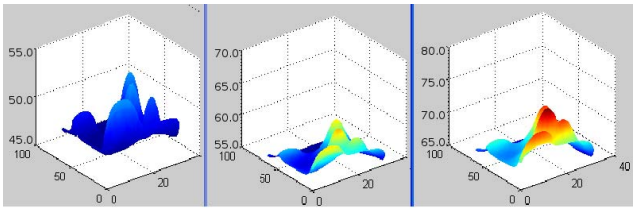
Figure 9 shows the evolution in time of the thermal map for the register file when the registers are accessed in a "chess board" manner. As can be seen, this access pattern homogenizes the temperature on the silicon because the accesses are distributed across a larger surface. Moreover, the probability of hotspots is minimized and the reliability of the system is not compromised. Therefore, this access policy can be considered as an effective way to optimize the thermal behavior of the shared register file from the compi-

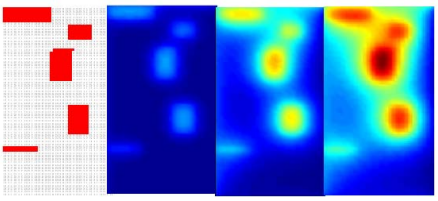












lation stage.

5 Experimental Section

Figure 10 shows the evolution in time of the temperature on the surface of the register file when running the application ADPCM_DECODE. For this set up, the register file has been implemented as an squared array of 64 registers.

Every one of the plots showed in the figure has been obtained every 5000 simulation cycles, and the benchmark has been run until completion. As can be seen, the profile on the register accesses causes that several registers are more demanded than others, creating, in this way, a thermal difference between registers (temperature gradient) that can destroy the device. In this example, the repetitive access to a few registers makes them to increase the temperature when compared with the whole set.

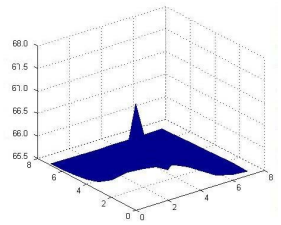
In order to evaluate the proposed mechanisms to reduce the thermal breakdown, the register assignment performed by the compiler is modified in a way that the registers from the register file are assigned in a "chess manner" policy. Figure 11 shows the new evolution in time of the temperature in the register file.

As can be seen, once the register assignment policy has been modified to assign the registers in such a special way, the thermal gradients disappear as the temperature is homogeneously distributed across the registers. The graph shows how the temperature of the register file slowly increases with the number of accesses, but the performed compiler modification avoids overused registers and, therefore, overheated active areas.

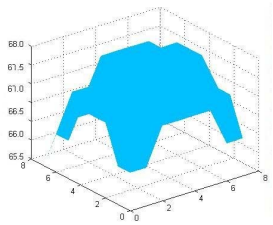
The same approach has been followed to mitigate the effect of thermal gradients when running an image processing algorithm (JPEG2000). The data presented in the plots are also acquired every 5000 simulation cycles. As can be seen in Figure 12, thermal gradients appear as a consequence of the different register usage. Also, the placement of the register file close to the hottest units in the system (cache memories) increments the temperature of the device by thermal diffusion.

The optimization of this example has been performed by modifying the register assignment phase as mentioned previously to diminish the temperature differences among the registers. Additionally, the placement of the register file has been selected to be close to the border area of the chip in order to avoid the thermal transfer from the hot cache memories and reduce the average temperature.

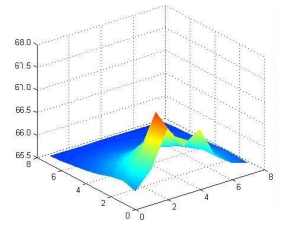
Figure 13 shows the evolution in time of the register file temperature when the register assignment policy has been



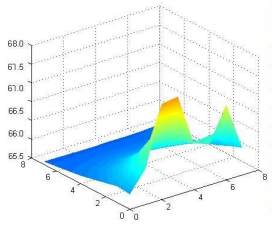
(a) First 5000 simulation cycles



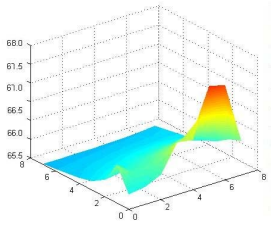
(b) Second 5000 simulation cycles



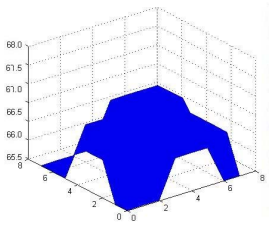
(c) Third 5000 simulation cycles



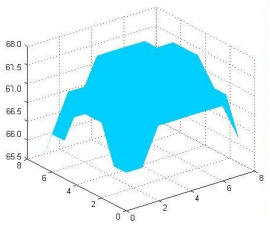
(d) Fourth 5000 simulation cycles



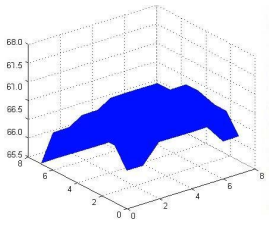
(e) Last simulation cycles



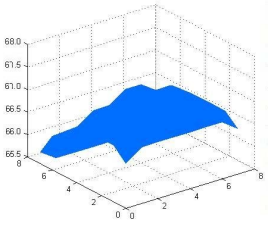
(a) First 5000 simulation cycles



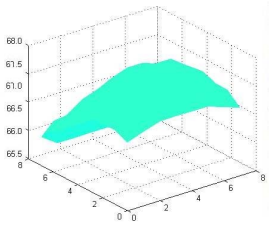
(b) Second 5000 simulation cycles



(c) Third 5000 simulation cycles



(d) Fourth 5000 simulation cycles



(e) Last simulation cycles

modified to perform a "chess based" assignment, as well as the placement of the register file has been moved to the border area. These graphs show how the temperature is homogenized for every register in the register file, and the final temperature of the device is also reduced.

6 Conclusions

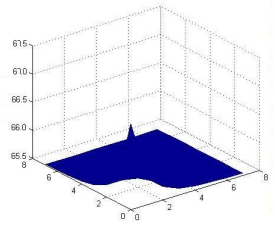
The thermal behavior of the hardware modules that integrate the architecture of the processor is a factor that must be controlled from different abstraction levels. For that purpose, accurate and flexible estimation mechanisms are required.

The work presented in this paper has shown the development of an analytical model to evaluate the thermal map of the register file, one of the hottest modules in the VLIW processor architectures. This model has been used to evaluate the effect of several high-level transformations in the register file placement, topology and register access pattern.

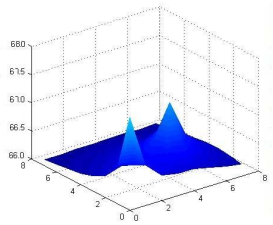
Finally, the presented model has demonstrated to be an effective mechanism to achieve the thermal map of the register file and has allowed to propose a register access policy that improves the thermal behavior.

Acknowledgements

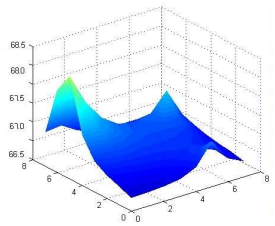
This work is partially supported by the Swiss FNS Research Grant 20021-109450/1, and the Spanish Government Research Grants TIN2005-05619 and TIC2003-07036.



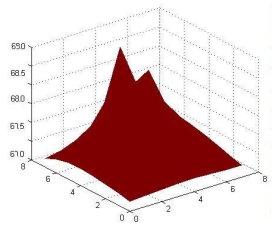
(a) First 5000 simulation cycles



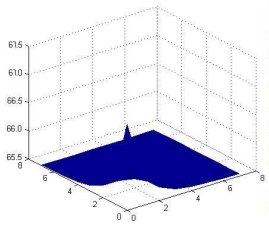
(b) Second 5000 simulation cycles



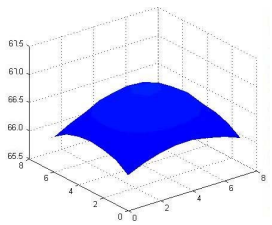
(c) Third 5000 simulation cycles



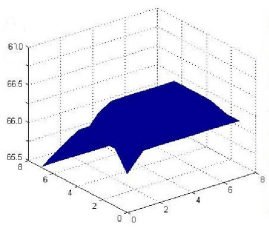
(d) Last simulation cycles



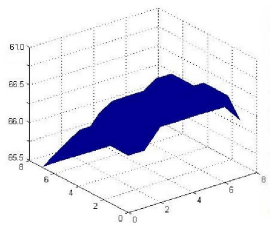
(a) First 5000 simulation cycles



(b) Second 5000 simulation cycles



(c) Third 5000 simulation cycles



(d) Last simulation cycles

References

[1] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *HPCA*, 2001.

[2] J. Donald and M. Martonosi, "Temperature-Aware Design Issues for SMT and CMP Architectures," in *Workshop on Complexity-Effective Design*, 2004.

[3] http://www.hpl.hp.com/research/dca/smart_cooling/.

[4] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full leakage estimation considering power supply and temperature variations," in *ISLPED*, 2003.

[5] P. Li, L. Pileggi, M. Ashegi, and R. Chandra, "Efficient full-chip thermal modeling and analysis," in *ICCAD*, 2004.

[6] W. Huang, E. Humenay, K. Skadron, and M. Stan, "The need for a full-chip and package thermal model for thermally optimized IC designs," in *ISLPED*, 2005.

[7] W. Huang, M. Stan, and K. Skadron, "Parameterized physical compact thermal modeling," *IEEE Trans. on Component Packaging and Manufacturing Technology*, vol. 28, no. 4, pp. 615–622, December 2005.

[8] S. Lopez-Buedo, J. Garrido, and E. I. Boemo, "Dynamically inserting, operating, and eliminating thermal sensors of FPGA-based systems," *IEEE Trans. on Components and Packaging Technologies*, vol. 25, no. 4, pp. 561–566, December 2002.

[9] N. Julien, J. Laurent, E. Senn, and E. Martin, "Power consumption modeling and characterization of the ti c6201," *IEEE Micro*, vol. 23, no. 5, pp. 40–49, September 2003.

[10] E. Senn, J. Laurent, N. Julien, and E. Martin, "Softexplorer: Estimation, characterization, and optimization of the power and energy consumption at the algorithmic level," in *International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2004.

[11] J. L. Ayala, C. Méndez, and M. López-Vallejo, "Analysis of the Thermal Impact of Source-Code Transformations in Embedded Processors," in *ICECS*, 2006.

[12] C. Méndez, J. L. Ayala, and M. López-Vallejo, "Target Independent Thermal Modeling for Embedded Processors," in *IES*, 2006.

[13] G. Paci, P. Marchal, F. Polletti, and L. Benini, "Exploring Temperature Aware Design in Low-Power MPSoCs," in *DATE*, 2006.

[14] D. Atienza, P. G. D. Valle, G. Paci, and F. Poletti, "A Fast HWSW FPGA-Based Thermal Emulation Framework for Multi-Processor System-on-Chip," in *DAC*, 2006.

[15] D. Atienza, P. Raghavan, J. L. Ayala, G. de Micheli, F. Catthoor, D. Verkest, and M. López-Vallejo, "Compiler-driven leakage energy reduction in banked register files," in *International Workshop on Power and Timing Modeling, Optimization and Simulation*, 2006.