Exploring "temperature-aware" design in low-power MPSoCs

G. Paci
DEIS, U. Bologna
gpaci@deis.unibo.it

P. Marchal
IMEC
marchal@imec.be

F. Poletti
DEIS, U. Bologna
fpoletti@deis.unibo.it

L. Benini
DEIS, U. Bologna
lbenini@deis.unibo.it

Abstract

The power density inside high performance systems continues to rise with every process technology generation, thereby increasing the operating temperature and creating "hot spots" on the die. As a result, the performance, reliability and power consumption of the system degrade. To avoid these "hot spots", "temperature-aware" design has become a must. For low-power embedded systems though, it is not clear whether similar thermal problems occur. These systems have very different characteristics from the high performance ones: they consume hundred times less power, they are based on a multi-processor architecture with lots of embedded memory and rely on cheap packaging solutions. In this paper, we investigate the need for temperature-aware design in a low-power systems-on-a-chip and provide guidlines to delimit the conditions for which temperature-aware design is needed.

1. Introduction

In recent years, the power densities in high performance microprocessors have doubled every three years [2]. Moreover, most power is consumed in a few localized spots which heat up much faster than the rest of the chip (e.g., the CELL processor [4]). These hot spots potentially increase leakage currents, cause timing errors and/or even result in physical damage. The heating has also become a big issue because expensive cooling solutions are not acceptable for consumer products. Several authors have therefore advocated the need for "temperature-aware" design techniques (see [18] for an overview). Some of them have already found their way in industrial designs (E.g., in the Intel Itanium processors [15]).

It is less clear whether such techniques are required for low-power multi-processors systems-on-a-chip (LP-MPSoC). These systems dissipate two orders of magnitude less heat (max 3W instead of 100W). Since portable systems are the main target for LP-MPSoCs (such as mobile phones), they are built in a low standby power technology instead of a high performance one. The main difference is that subthreshold leakage is engineered to remain low, even at the expense of a higher power supply and thus more dynamic power (see table 1). Due to the smaller contribution of leakage power, the impact of temperature on the total power dissipation is limited [9]. However, packaging solutions for consumer electronics are much cheaper and rely on natural convection for removing the heat. As a result, the die is thermally more isolated and may still heat up.

The existence of hot spots on the die is even harder to predict. LP-MPSoCs for instance are built with a different computer architec-

65nm technologies	Vdd(V)	Vt(V)	Ioff(uA/um)
high performance	0.9	0.18	7e-2
low power	0.8	0.26	5e-3
low standby power	1.1	0.5	2.5e-5

Table 1: Technology bifurcation in the 2003 ITRS roadmap. Leakage is significantly lower in low standby power than high performance technologies.

ture than high performance systems. E.g., multiple power-efficient processors instead of a single complex super-scalar processor (compare a Intel P4 with a Philips Nexperia). Secondly, they operate at a lower power density: the processors run at a lower speed (typically around 500Mhz instead of 3GHz) and contain a large amount of embedded memory. Finally, the die of a LP-MPSoC is typically smaller than that of high performance computer. Therefore, the generated heat can more easily reach the corners of the chip, resulting in less temperature variations across the die. Several factors thus seem to indicate that hot spots are less likely on LP-MPSoCs than on high performance systems. However, this has to be verified.

The contribution of this paper is *to delimit the conditions for which hot spots become a critical problem in LP-MPSoCs*. To investigate this problem, we have built an accurate thermal/power model of a multi-processor system-on-a-chip. Our thermal model is different from those of high performance systems because we investigate the thermal behavior of multiple cores and embedded memories on a single die and we look at package solutions for LP-MPSoCs which have a much higher thermal resistance.

For modeling the heat flow, we rely on an equivalent electrical RC model (similar to HotSpots [18]) which we have calibrated against a 3D-finite element analysis. To obtain realistic trace of the activities on the die, we have integrated our thermal model in a multi-processor system-on-a-chip simulator. Experimental results for a typical LP-MPSoC show that the temperature differences on chip are limited and that the temperature changes only slowly with time.

This paper is organized as follows. First, we discuss the related work on thermal modeling and temperature-aware design (section 2). Then, we explain our target architecture (section 3) and how we have modeled its thermal behavior (section 4). Finally, we present our experimental results (section 5).

2. Related work

Three problems arise due to an elevated operating temperature. First, the higher the temperature becomes, the more leakage currents occur and thus the more power is consumed. This is researched

for instance by [19][21][11]. Secondly, a higher temperature also reduces the mean-time-to-failure of the system. For instance, the physical processes that trigger electromigration and stress migration become more active if temperature rises (see [14][12]). Thirdly, a higher temperature impacts the performance of the transistors. It on the hand reduces the mobility of the charge carriers (electron and holes), but on the other hand decreases the VT [9]. Usually, higher temperature decrease the performance of circuits. Timing violations then become more likely.

To investigate the above issues in detail, the authors of [18] have developed a thermal/power model for super-scalar architectures. It not only predicts the temperature variations between the different components of a processor, but also accounts for the increased leakage power and reduced performance. Their results clearly prove the importance of hot spots in high performance systems.

Based on this and/or similar models, many architectural extensions have been proposed to reduce the impact of hot spots and/or to prevent the die from breaching a critical temperature. The power density in super-scalar processors can be reduced with fetch toggling, decode throttling, frequency and/or voltage scaling (e.g., [1][7][17][5][15]). Except for frequency/voltage scaling, the above techniques are only applicable on super-scalar processors. Another approach for reducing the impact of hot spots is adding redundancy to the architecture. [18] advocates that a spare register-file and migrating computation between register files is the best in response to heating. Similarly, [6] examines the benefits of moving computation between multiple replicated units of a super-scalar processor.

Besides architectural solutions, the temperature can also be reduced at the system level. E.g., [16] stops scheduling hot tasks when the temperature reaches a critical level. In this way the system is idling and the CPU spends more time in low-power states, such that the temperature either locally or globally is decreased.

The related work discussed above is targeting high performance systems, where the power density hampers scaling. However, the context is different in low-power systems, which run at a lower speed and are subjected to less leakage power. [19] have investigated the impact of temperature and voltage variations across the die of an embedded core. Their results show that the temperature varies around 13.6 degrees across the die. Since they use a 130nm CMOS Silicon-on-Insulator technology, it is hard to extrapolate their results to a bulk CMOS technology. [8] explains a allocation and scheduling algorithm for eliminating hot-spots in a system-on-chip, but they target a much higher power budget (up to 15W) than is present in systems-on-a-chip for portable applications (up to 3W see description of the system drivers in the ITRS roadmap).

Based on our literature study, we believe that the case for temperature-aware design has not yet been made for low-power multi-processor systems. Particularly, the existence of hot spots in these systems has not been validated yet. In the remainder of this paper, we investigate this problem.

3. A typical LP-MPSoC

In figure 1, we show the floorplan of a typical LP-MPSoC. It consists of 16 ARM7 cores and 16 32kB shared memories. The shared memory is used for storing large data structures and communicat-

ing data between the processors. Each ARM7 core is attached to a local 8kB 2-way associative data cache and a 8kB direct mapped instruction cache. The memories and processors are connected using a XPipes Network-on-Chip [3] of which a 6x6 switch and network interface modules are shown on the floorplan. We have obtained the dimensions of the NoC circuits by synthesizing and building a layout. The dimensions of the memories and processors are based on numbers provided by an industrial partner. In the remainder of the paper, we will use this floorplan to research the temperature effects inside an LP-MPSoC.
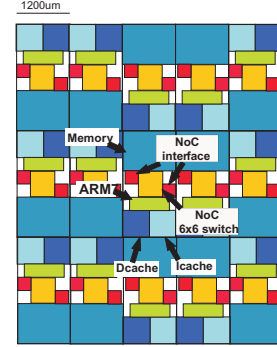

Figure 1: Floorplan of a multi-processor system on a chip in a 130nm technology:16 ARM7 processors each connected to a 2-way associative 8kB data cache and a direct mapped 8kB instruction cache; 16 memory tiles of 32kB; a NoC connecting processors and memories. Thermal cells are *150umx150um* (subsection 4.2.1).

4. Power/thermal analysis

To estimate the power consumption and temperature of all architectural blocks inside an LP-MPSoC, we have built a simulation environment as depicted in figure 2.
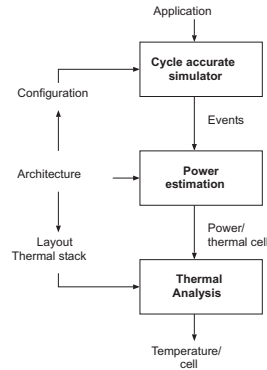

Figure 2: Power/thermal model integrated in a cycle-accurate LP-MPSoC simulator [13]

We use a cycle accurate simulation platform for measuring the activities in each of the memories and processing elements in the system[1]. We measure the time that each processor spends in active/stalled/idle mode. We trace the number and type of accesses to each of the memories (instruction cache/data cache and large shared

---

1   based on MPARM [13].

memories). Based on these activities, we estimate every $0.1us$ the energy consumption in each of the thermal cells of the layout. Thereafter, we feed this data into our thermal simulator. The latter one computes the temperature evolution of the die during the last $0.1us$. The temperature map of the die is then logged in a file. Since the joint performance/power/thermal simulation is rather time-consuming, we also provide the option to dump the components' activities in a trace-file. Using this trace-file, we can more quickly explore different packaging solutions.[2]

In the next subsections, we first discuss the power model used during our experiments (subsection 4.1). Secondly, we explain the chip's thermal model in detail (subsection 4.2).

4.1. Power estimation

		Max. Power@100Mhz	Max. Power density
ARM7		5.5mW	$0.03W/mm^2$
DCache	8kB/2way	43mW	$0.012W/mm^2$
ICache	8kB/DM	11mW	$0.03W/mm^2$
Memory	32kB	15mW	$0.02W/mm^2$

Table 2: Power for the most important components of a LP-MPSoC in a 130nm bulk CMOS technology.

In Table 2, we outline the power consumption and power densities of the most important components of our system-on-a-chip. The table contains the maximum power numbers, but the effective power is normally lower, depending on the workload (activities of processors and memories). We ignore leakage energy. Leakage in mobile systems has to be limited for guaranteeing sufficient battery-life time. Typically, leakage is therefore eliminated at the device level by developing high Vt transistors. As a result, the leakage can be as low as $25pA/um$ (see ITRS roadmap). High Vt transistors come at the expense of a higher Vdd and thus more dynamic energy (since the Vdd-Vt has to be kept constant for retaining the same performance). A better option is to use leakage reduction techniques such as back-biasing. E.g., [10] jointly optimize Vdd/Vt by using dual-gate devices or back-biasing. The authors lower the Vdd for reducing the dynamic power. However, to retain the same performance, they have to reduce Vt as well, which unfortunately exponentially increases the subthreshold leakage. In the optimal Vdd/Vt operating point (where the Vdd/Vt are more aggressively scaled than on the ITRS roadmap[3]), leakage energy contributes only 10% of the total power. Hence, we believe that the impact of leakage on temperature for low-standby power systems is limited.

In the next section, we discuss the flow of the dissipated power through the die.
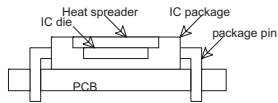
4.2. Modeling the heat flow

Figure 3: Chip packaging solution

---

2   In this case though, we cannot feedback the temperature effects to the performance simulator

3   Hence, the system operating with the optimal Vdd/VT is more leaky than with ITRS conditions.

An LP-MPSoC is usually packed within a cheap plastic ball grid array package ([20] and see figure. 3). The main purpose of this package is to electrically connect the die to the other circuits on the printed circuit board and to protect it against the environment. Besides, the package has also to remove the dissipated heat from the die. Typically, a heat spreader made of copper, alu or another highly conductive material is therefore attached to the reverse side of the die. Its goal is to increase the thermal conductivity of the package. In the context of this paper, we assume that all surfaces but the one of the heat spreader are adiabatic. The spreader disposes the generated heat by natural convection with the ambient.
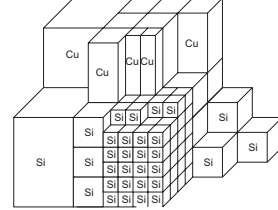

Figure 4: Dividing the chip into a finite number of cells

4.2.1. Equivalent RC thermal model Similar to [18][19][6], we exploit the well known analogy between electrical circuits and thermal models. We decompose the silicon die and heat spreader in elementary cells which have a cubic shape (figure 4) and use an equivalent RC model for computing the temperature of each cell. By varying the cell size we can trade-off the simulation speed of the thermal with its accuracy. The coarser the cells become, the less cells we need to simulate, but the less accurate the temperature estimates become. The cell sizes used during our experiments are $150um*150um$ (see figure 1). We assume that the power uniformily burned in this region (which is 1/8th of the size of an ARM processor in 130nm). For technologies which have a worse thermal conductance (such as fully depleted SOI), we plan to use smaller thermal cells (down to the level of standard cells).
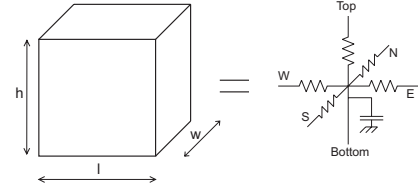

Figure 5: Equivalent RC circuit of a cell

We associate with each cell a thermal capacitance and five thermal resistances (figure 5). Four resistances are used for modeling the horizontal thermal spreading whereas the fifth is used for the vertical thermal behavior. The thermal conductivities and capacitance of the cell are computed as follows (where $k_{th}^{si/cu}$ is the thermal conductivity and $c_{th}^{si/cu}$ is the thermal capacitance per unit volume):

$$G_{th}^{NESW} = k_{th}^{si/cu} \cdot \frac{h \cdot w}{l} \tag{1}$$

$$G_{th}^{top} = k_{th}^{si/cu} \cdot \frac{l \cdot w}{h} \tag{2}$$

$$C_{th} = c_{th} \cdot l \cdot h \cdot w \tag{3}$$
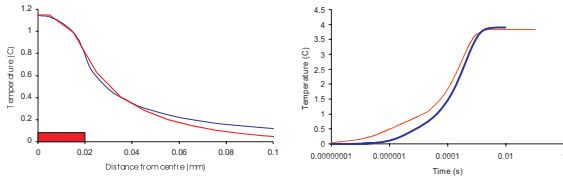
$$\tag{4}$$

Figure 6: Comparison of the spatial (left) and temporal temperature (right) distribution evaluated with 3D finite element package (light line) vs. equivalent RC model (bold line).

We model the generated heat by adding an equivalent current source to the cells on the bottom surface. The heat injected by the current source into the cell corresponds to the power density of the architectural component covering the cell (e.g., a memory decoder or processor) multiplied with the surface area of the cell. Note that no heat is transferred down into the package from these bottom cells.

In contrast, the heat from the cells on the top surface is removed through convection. We model this by connecting an extra resistance in series with their $R_{th}^{top} = 1/G_{th}^{top}$ resistance. The value of this resistance is equal to the package-to-air resistance weighted with the relative area of the cell to the area of the spreader.

4.2.2. Thermal properties In table 3, we enumerate the thermal properties of the package used during our experiments. The amount of heat that can be removed by natural convection strongly depends on the environment (such as the placement of the chip on the PCB, the case of the embedded system, etc.). A good average value is 20W/K (see [20]), even though this is much higher than the ones published by package vendors.

silicon thermal conductivity[4]	$150 \cdot \left(\frac{300}{T}\right)^{4/3} W/mK$
silicon specific heat	$1.628e - 12 J/um^3 K$
silicon thickness	$350um$
copper thermal conductivity	$400 W/mK$
copper specific heat	$3.55e - 12 J/um^3 K$
copper thickness	$1000um$
package-to-air conductivity	$20K/W$ in low power

Table 3: Thermal properties

Hot spot dimension(μm^2)	Temp 3D	Temp RC model
60×70	$2.89°K$	$2.83°K$
40×50	$2.39°K$	$2.37°K$
30×30	$2.02°K$	$2.08°K$

Table 4: Comparison of the maximum temperature reached with a 3D model and our equivalent RC model

4.2.3. Model calibration We have compared and calibrated our thermal model with a 3D-finite element package. For this purpose, we have modeled a single heat source located in the center of the chip's bottom surface. The temperature of this source as predicted by the 3D model and our RC model is shown in table 4.

Besides predicting the steady state temperature within the hot spot, we also validate the our model's prediction of the spatial distribution of the temperature. In figure 6-left, we illustrate how the temperature decreases in function of the radial distance from the center of the heat source. The size of source is indicated by the grey box. The predictions of both models are again similar.[5] Finally, we
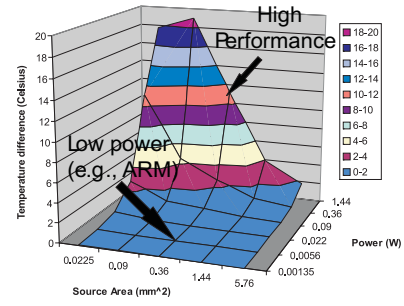

Figure 7: Temperature differences on chip (6000umx7200um). It contains in the center a single source of which the area and power is varied.

test the accurateness of our thermal model for predicting the temporal behavior of the die. We apply a sudden power load to the center of the chip and illustrate the temperature response of the die in figure 6-right.

In the next section, we discuss experimental results which we have obtained with this simulation environment.

5. Experimental results

5.1. Thermal properties of the die

To delimit the conditions when larger temperature differences on the die of the MPSOC will occur, an experiment with a single heat source was conducted of which the size and power is varied. The resulting maximum temperature differences on the chip are shown in figure 7.

For a given area of the power source (e.g., $0.36mm^2$), the temperature differences increases proportional to the power (with 7.7K/W). For a given power budget (e.g., $0.36W$), the temperature is proportional to the inverse of the square root of the area (with 1.8Kmm). The smaller the power source becomes, the smaller the surface becomes through which the heat can be removed. As a result, the thermal resistances increases with a decreasing diameter of the power source and the largest temperature drop occurs near the power source. The thermal resistance is thus mainly determined by the area of the source rather than the distance from the source to the coldest point on the die. Hence, even if a larger die size is used, the temperature differences will not increase.

With this figure, designers can easily predict the temperature differences that will occur on their die and thus delimit the conditions for which thermal design is required. E.g., in our low-power embedded systems the area of the processor is around $0.3mm^2$ and consumes $5mW$. As can be seen in this graph (and further illustrated in the next section), the resulting hot spot will be very low. High performance systems operate in a different field: they consume much more power for the same area. This is mainly because they operate at a higher clock frequency (e.g., 30 times faster = 30 times more power). For achieving this high clock frequencies, they use a high performance technology in which the leakage contribution cannot be ne-

---

5 At large distances from the center of the heat source, our RC model underestimates the temperature, because away from the source, we have used larger cell sizes for reducing its run time.

glected (e.g., leakage is as important as dynamic energy = 2 times more power). Moreover, they use a different circuit style (such as dynamic logic vs. static logic), that is more power hungry (e.g., 2 times more power) and rely on more complex IP blocks (such as complex multiport register-files). As a result, they consume 100 times more power on the same area, reaching power densities larger than $1W/mm^2$. This results important hot spots on the die (12 degrees for our die).

To further validate our results for low power MPSoCs, we look in the thermal issues with more precise thermal/power simulations.

5.2. Steady-state thermal analysis

In figure 8-left, we show the temperature differences estimated on the die when running a pipelined matrix multiplication on four processors. Matrix multiplications forms the core of most multi-media algorithms (DCT,Wavelets, etc.). It is a very compute and data intensive application (and thus power hungry). The hottest parts of the chip are the processor cores and their instruction memories, since they are most active and thus dissipate the most power. They are followed by the data cache. Even though that the data cache consumes more energy per access than the instruction cache, it is less actively used[6] and therefore does not become a hot spot. When running at 100Mhz, the temperature differences on the chip are limited (max. 0.128 Celsius). The on-chip temperature difference increases only slightly when six additional processors are started (see figure 8-middle: max. 0.142 Celsius). This is because the processors with a relatively high power density are intermingled with memories which a much lower power density.

By scaling the frequency of the processors from 100Mhz to 1GHz, the power increases with a factor 10. As a result and in agreement with the results of figure 7, the temperature differences on the chip increase with a factor 10. We have measured a temperature difference of 1.53 Celsius.

In a next experiment, we test the impact of technology scaling. We scale all the dimensions of our layout with a factor 2 (reflecting a technology change from 130nm to 65nm). We also assume that the power supply does not scale in future processing generations (which is a worst-case assumption) and use a clock of 1Ghz (which is high for low power systems). According to figure 7, we find that scaling the area of the power source by four, results in 1.8 times larger temperature differences on chip. This estimation is confirmed by our accurate simulation: temperature gradients increase from 1.5 Celsius (130nm@1Ghz) up to 2.8 Celsius (65nm@1Ghz) (see figure 8-right).

From these results, we conclude that the on-chip temperature differences of a typical LP-MPSoC are limited. More important, figure 7 allows to easily predict the on-chip temperature differences.

5.3. Transient thermal analysis

So far, we have only considered spatial temperature differences. However, temporal temperature differences or thermal cycles are equally important (e.g., they impact reliability). We therefore plot the average die temperature in function of the time (see figure 9-left).

As the thermal resistance of the die with the environment is rather high (20K/W), it takes around 8s before the steady state temperature is reached. The resulting temperature depends on the power burned on the chip: the more processors running, the higher the final temperature becomes (see the differences between 4 and 10 cores in the figure). In the figure 9-middle, we plot the maximum temperature difference in function of the time. The steady state temperature difference is reached after only 250ms. This is much faster than the temperature equilibrium of the die with the environment. It can be explained by the fact that the silicon die and the cu heat spreader are good thermal conductors. Since the thermal time constant on the die is low, the on-chip temperature differences may be very sensitive to variations of the power consumptions, i.e. thermal cycling. To analyze thermal cycles due to a varying workload, we have generated an artificial benchmark application running on a single processor. It consists of a period of high activity followed by one of low activity. Its power and temperature profile are shown in figure 9-right. Globally, the temperature increases as the steady state temperature of the die has not been reached yet.[7] A series of thermal cycles is superposed on this gradual increase of temperature. Their amplitude is very small as the thermal resistance of the die is small.

Large thermal cycles are mainly due to the high resistance of the package with the environment and only occur at a large time scale. Therefore, we conclude that thermal cycling (and the resulting reliability issues [14]) are less of a problem in LP-MPSoCs than in high performance systems.

6. Conclusions

In this paper, we have investigated the need for temperature-aware design in LP-MPSoCs. We have built a thermal model which we have calibrated with a 3D finite-element analysis. Based on our experimental results for a typical LP-MPSoC, we observe that no hot spots occur across the die. In the context of LP-MPSoCs implemented on bulk CMOS and under the plausible assumption that LP-MPSoCs will not rush for super-fast clocks (such as defined in the ITRS roadmap for high performance logic), we therefore do not see the immediate need for techniques to analyse and reduce hot spots. However, if more advanced packaging solutions (such as 3D stacking) and new low-k dielectrics are introduced in the BEOL, the thermal model of the chip may fundamentally change. The presence of hot spots in these novel technologies has to be investigated for LP-MPSoCs. Furthermore, as the steady-state temperature depends on the packaging solution and the applied workload, temperature-aware design remains necessary to assure that the maximal temperature of the system is not breached.

References

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

[2] D.Frank, R. Dennard, E. Nowak, P. Solomon, Y. Taur, and H. Wong. Device Scaling Limits of Si MOSFETs and Their Application Dependencies. *Proc. IEEE*, 89(3):259–, Mar. 2001.

---

6   The register file acts as an extra level of cache and thus eliminates accesses to the data cache

7   The measurement is done after 5s, but the steady state temperature is only reached after 8s.
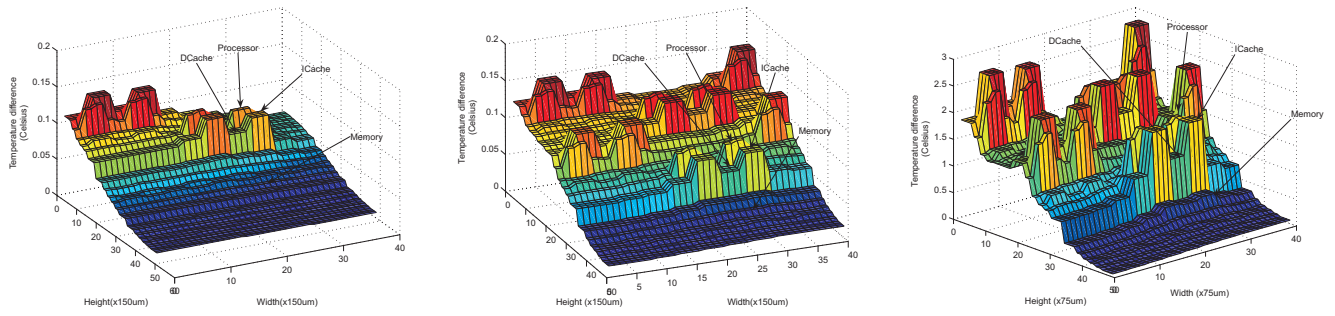
Figure 8: Temperature differences of a multi-processor system-on-a-chip. (left) 4 processors running at 100MHz; (middle) 10 processors at 100Mhz; (right) 10 processors at 1GHz on a scaled die size.
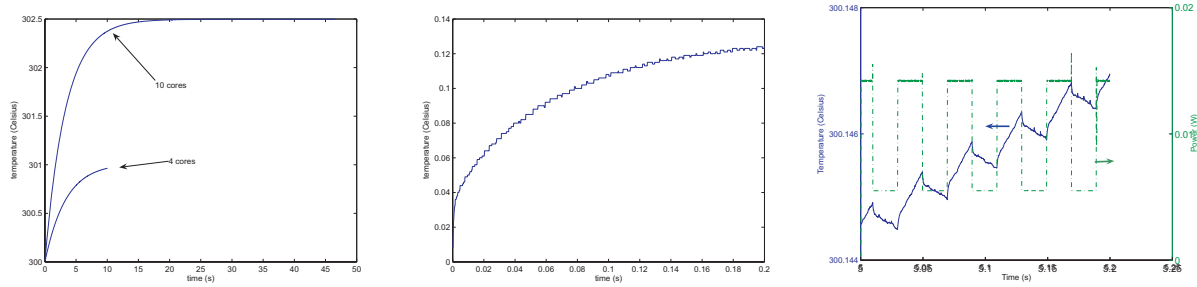

Figure 9: Temporal behavior of the die: (left) average die temperature; (middle) temperature differences on chip; (right) thermal cycles on chip

[3] D. Bertozzi et al. NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip. *IEEE T. Parallel and Distributed Systems*, 16(2):113–129, 2005.

[4] D. Pham et al. The design and implementation of a first-generation CELL processor. In *Proc. IEEE/ACM ISSCC*, pages 184–186, Feb. 2005.

[5] S. Gunther, F. Binns, and D. Carmean and. J. Hall. Managing the impact of increasing microprocessor power consumption. *Intel Technology Journal Q1*, 2001.

[6] S. Heo, K. Barr, and Krste Asanovic. "Reducing Power Density through Activity Migration. In *Proc. IEEE/ACM ISLPED*, Aug. 2003.

[7] M. Huang, J. Renau, S. Yoo, and J. Torrellas. A framework for dynamic energy efficiency and temperature management. In *Proc. ACM/IEEE Micro*, pages 202–213, Dec. 2000.

[8] W. Hung, Y. Xie, N. Vijaykrishnan, M. Kandemir, and M Irwin. "Thermal-aware allocation and schedulilng for systems-on-chip design. In *Proc. IEEE/ACM DATE*, pages 898–899, Feb. 2005.

[9] K. Kanda, K. Nose, H. Kawaguchi, S. Lee, and T. Sakurai. Design Impact of Positive Temperature Dependences on drain current in Sub 1V CMOS VLSIs. *IEEE J. Solid State Circuits*, 36(10):1559–, Oct. 2001.

[10] J. Kao, M. Miyazaki, and A. Chandrakasan. A 175mV Multiple-Accumulate Unit Using an Adaptive Supply Voltage and Body Bias Architecture. *IEEE J. Solid-State Circuits*, 37(11):1545–1555, Nov. 2002.

[11] W. Liao, F. Li, and L. He. "Microarchitecture level power and thermal simulation considering temperature dependent leakage model. In *Proc. IEEE/ACM ISLPED*, pages 211–216, Aug. 2003.

[12] Z. Lu, W. Huang, J. Lach, M. Stan, and K. Skadron. "Interconnect Lifetime Prediction under Dynamic Stress for Reliability-Aware Design. In *Proc. IEEE/ACM ICCAD*, Nov. 2004.

[13] Loghi M., Angiolini F., Bertozzi D., Benini L., and Zafalon R. Analyzing Chip Communication in a MPSoC Environment. In *Proc. IEEE/ACM DATE*, Feb. 2004.

[14] J. Person. Scaling-Induced Reductions in CMOS Reliability Margins and the Escalating Need for Increased Design-In Reliability Efforts. In *Proc. ISQED*, pages 123–, 2001.

[15] C. Poirier, R. McGowen, C. Bostak, and S. Naffziger. Power and temperature control on a 90nm Itanium-Family processor. In *Proc. IEEE/ACM ISSCC*, pages 304–305, Feb. 2005.

[16] E. Rohou and M. Smith. "Dynamically managing processor temperature and power. In *Proc. 2th workshop on Feedback-directed optimization*, Nov. 1999.

[17] H. Sanchez, B. Kuttanna, T. Olson, M. Alexander, G. Gerosa, R. Philip, and J. Alvarez. "Thermal management for high performance PowerPC microprocessors. In *Proc. IEEE Compcon*, 1997.

[18] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. "Temperature-Aware Microarchitecture. In *Proc. IEEE/ACM ISCA*, pages 2–13, Jun. 2003.

[19] H. Su, F. Liu, A. Devgan., E. Acar, and S. Nassif. "Full chip leakage estimation considering power supply and temperature variations. In *Proc. IEEE/ACM ISLPED*, pages 78–, Aug. 2003.

[20] B. Vandevelde, E. Driessens, A. Chandrasekhar, and E. Beyne. Characterisation of the polymer stud grid array (PSGA), A lead free CSP for high performance and high reliable packaging. In *Proc. SMTA*, Sept. 2001.

[21] Y. Zhang, D. Parikh, K. Sankaranarayanan, K. Skadron, and M. Stan. HotLeakage: A Temperature-Aware Model of Sub-threshold and Gate Leakage for Architects. Technical Report CS-2003-05, Univ. of Virginia Dept. of Computer Science, Mar. 2003.