A Simulation Methodology for Reliability Analysis in Multi-Core SoCs

Ayse K. Coskun, Tajana Simunic Rosing
University of California San Diego (UCSD)
9500 Gilman Dr.
La Jolla CA 92093-0404
{acoskun, tajana}@cs.ucsd.edu

Yusuf Leblebici, Giovanni De Micheli
Ecole Polytechnique Federale de Lausanne
(EPFL), CH-1015, Switzerland
{yusuf.leblebici, giovanni.demicheli}@epfl.ch

ABSTRACT

Reliability has become a significant challenge for system design in new process technologies. Higher integration levels dramatically increase power densities, which leads to higher temperature and adverse effects on reliability. In this paper, we introduce a simulation methodology to analyze reliability of multi-core SoCs. The proposed simulator is the first to provide system-on-chip level fine-grained reliability analysis. We use our simulation methodology to study the reliability effects of design choices such as thermal packaging and placement, as well as runtime events such as power management policies and workload distributions.

Categories and Subject Descriptors: B.8.0 [Performance and Reliability]: General; C.4 [Performance of Systems]: Modeling Techniques.

General Terms: Reliability, Measurement.

Keywords: MP-SoC reliability, reliability simulation, reliability modeling.

1. INTRODUCTION

Reliability is becoming a limiting factor in system-on-chip designs due to the high failure rates in deep submicron and nanoscale devices. The increase in failure rates is caused by high integration levels, higher power and temperature densities and scaling of transistor dimensions. The rate of hard faults occurring in useful life of devices is tightly coupled with the increasing temperature. Thus, the problems of high power consumption and temperature not only affect the design cost due to the need for sophisticated cooling solutions, but also raise significant challenges for system reliability.

Dynamic power management (DPM) ([2]) and dynamic voltage scaling (DVS)([10]) have been proposed to reduce power consumption. Lower power consumption reduces the average temperature on chip, typically resulting in a positive effect on reliability as the rate of temperature induced faults is decreased. However, aggressive power management can adversely affect reliability by increasing the rate of hard faults due to temperature cycling phenomena [15]. Power management techniques do not always succeed in eliminating the hot spots when the system is highly utilized. In addition, designing the thermal package for the worst-case power dissipation is prohibitively expensive. Dynamic Thermal Management (DTM) addresses these issues and takes precautions for hot spots [18]. System reliability benefits from DTM as the hot spots on chip are avoided. DTM makes sure critical temperatures are not exceeded on chip, but it does not consider the temperature variations that result in temperature cycles. Therefore, reliability modeling and management are needed to address the temperature induced reliability problems on SoCs effectively.

While DPM, DVS and DTM have been widely investigated in literature, there has been limited work on analyzing system reliability. RAMP [19] is the first architecture level tool for modeling the reliability of microarchitectural units. In [15], system reliability is evaluated using a high level statistical model.

In this paper, we introduce a simulation methodology for multi-core SoC reliability analysis. Statistical models are at a high abstraction level, and therefore fine-grained monitoring of system behavior is not possible. Architecture level models typically sample temperature changes on the order of microseconds in order to accurately monitor workload changes in each unit, and thus are too detailed and time consuming to be applicable to large multi-core SoCs. Our methodology fills the gap between statistical and architecture level models for analyzing MPSoC reliability at core level. Temperature induced hard failure mechanisms are influenced by a number of factors related to both design choices and runtime events. We analyze thermal packages and placement of cores on the chip as design choices, and we emphasize the reliability challenges for low-cost packages. Runtime events we investigate are various power management strategies and workload distributions. The rest of the paper begins with related work in Section 2. In Section 3, we explain the simulation methodology. Section 4 presents the results on reliability analysis and Section 5 concludes.

2. RELATED WORK

Thermal modeling and management have become crucial in new technologies. HotSpot ([18], [4]) is a compact thermal model developed at architectural level, which allows tracking temperature changes on each unit. Dynamic thermal management (DTM) has been proposed to prevent the thermal

GLSVLSI'06, April 30–May 2, 2006, Philadelphia, PA, USA.
Copyright 2006 ACM 1-59593-347-6/06/0004 ...$5.00.

hot spots on chip and system failures due to over-heating. In [20], a predictive DTM approach is introduced that predicts the hot spots based on the profiling information of multimedia benchmarks. Response mechanisms such as resizing the instruction window are shown to have less impact on performance in comparison to reactive strategies such as fetch-toggling proposed in [16].

Srinivasan et al. developed a model for reliability at architecture level (RAMP) for intrinsic hard failures [19]. RAMP focuses on the effects of application behavior on reliability, and optimizes the architectural configuration and DTM policies for increased reliability. Exploiting architecture's knowledge of the runtime workload distribution, the proposed DTM methodology keeps the highest temperature reached on the processor under a given threshold by architectural adaptation and dynamic voltage scaling (DVS). In [14], system-on-chip is modeled by using a power and reliability state machine, and the effects of DPM on reliability are investigated. [15] points out the power and reliability management trade offs on SoCs, and proposes a joint policy optimization method that achieves power savings up to 40% while meeting the reliability criteria.

In this paper, we analyze the reliability of multi-processor SoCs at system level, emphasizing the effects of design choices and runtime events, such as thermal packaging and power management policies. Failure rate variations at run-time can be observed using the proposed methodology. Run-time reliability analysis gives the opportunity to apply dynamic strategies to overcome the temperature related reliability challenges. Our model is at a fine-grained level in contrast to statistical models. It also differs from the architecture level models by its ability to sample the variations at a granularity similar to the activity level of power management policies. This provides an opportunity to analyze reliability in much longer time frames, while permitting accurate observation of temperature variations.

3. METHODOLOGY

The proposed simulation methodology is the first to enable fine-grained reliability analysis of multi-core SoCs due to temperature induced hard failure mechanisms. Large magnitudes of temperature variations (e.g. 10-20 Celsius) can occur even during several seconds of processor execution [18, 17], motivating fine grained analysis. Architecture level models accurately analyze the temperature related problems on a processor. However, architecture level modeling is typically at the granularity level of microseconds for capturing the activity precisely at each unit. Such a detailed model is not feasible for large SoCs with many cores due to long simulation time and complexity. Moreover, SoC management strategies such as DPM and DVS make decisions at much longer time periods than the execution time of several instructions.

Our fine-grained simulation gives the opportunity for observing run-time temperature and failure rate variations. The simulation framework starts with generation of workload. The methodology is event-driven, where at each workload arrival or finish, power manager is invoked. Depending on the policy, power manager's decision can alter the power state and result in significant temperature changes. So, temperature is sampled at each workload event and the corresponding failure rate is computed. This method provides a similar sampling granularity with the decisions of power management policies. This way, we are able to observe changes in failure rate at a fine-grained level. At the same time, the simulation is much faster than architecture level models.

We next provide details for each part. We describe specific approaches for modeling and simulation; however, each part can be replaced with another compatible implementation.

3.1 Workload Model for SoC

In workload generation, our goal is to obtain workload statistics for each processing unit in the system, demonstrating the active and idle phases. Workload statistics can be obtained by using a multi-core power/performance simulator or emulator, or by creating synthetic workloads. An example multi-core cycle-accurate simulator for ARM7 cores is MP-ARM [1]. Multi-core simulation/emulation platforms gather real workload information for selected applications. Using synthetic workload enables generation of a variety of workload scenarios. In this work, synthetic workloads are chosen in order to perform a generalized study of relations between workload, power management and reliability.

To obtain workload statistics, we create several task sets using random task generation method, which is commonly used in literature [11]. We assume independent periodic tasks. Based on real life tasks ([8]), we assign an equal probability to each task for having a short (50-200ms) or long period (200-1000ms), similar to the approach in [10]. Each task is given a random WCET (worst case execution time) in the range (1-100ms). Since task scheduling is not the focus of this work, we do not consider dependent tasks or task graphs. The tasks are distributed to cores using a global earliest-deadline-first (EDF) schedule. The number of tasks and workload distribution among cores vary in our simulations.

3.2 Power Management

The second step of the simulator is the *Power Manager*, where the DPM and DVS strategies are applied. Power manager gets the workload trace as input, and applies the policy. Various power management techniques can be integrated within the simulator. In this work, we implement a fixed timeout DPM policy ([2]), in which a core is transitioned into sleep state only if it is idle for at least the breakeven time (T_{be}). T_{be} is the minimum time the core needs to stay in sleep state in order to save power. In our simulations, we take T_{be} as the total transition time of the processor into and out of the sleep state (T_o). Karlin's policy [6], which sets T_{be} as the fixed timeout, guarantees the power consumption to be at worst twice the amount of power consumed by an oracle policy with perfect workload information.

DVS policies typically make a decision on the next frequency/voltage setting based on the utilization ratio in an observation window of the core's past. In this work, we use a strategy derived from the utilization updating technique [7]. This technique estimates required processor performance by recalculating the utilization at each scheduling point. In our DVS policy, the processor utilization is computed for the particular core at each workload arrival. Based on the past utilization of the core, the target frequency is calculated. Among the available frequency settings, the frequency that is closest to the target (equal or higher than the target) is selected. This methodology is simple to implement for real systems and reaches reasonable power savings.

3.3 Thermal Modeling

Detailed thermal modeling has become a requirement for both thermal and reliability management of systems as designing for the worst case is getting prohibitively expensive. Thermal models are based on constructing an equivalent RC network. In this network, heat flow is analogous to the current passing through a thermal resistance whereas the transient behavior of temperature is modeled by means of the thermal capacitance. Compact thermal modeling tools have been proposed to address the need for detailed thermal analysis (e.g. [18]).

We use HotSpot version 2 [18] as the thermal modeling tool. HotSpot models the vertical and horizontal thermal resistances and capacitances automatically when the dimensions and material properties of units and chip are provided. It includes a very detailed thermal package model as well for a typical package set-up for today's chips. The thermal simulator takes into account the lateral thermal diffusion on chip, which increases its accuracy.

3.4 Failure Rate Modeling

We consider failures occurring in useful lifetime of devices, which are modeled with exponential distributions. In addition to obtaining average mean-time-to-failure of the system, we observe the failure rate variations over time at each core. Peaks in the failure rate are significant, since periods of very high failure rates increase the probability of a failure in that interval.

We model temperature induced intrinsic hard failures, which occur during processor lifetime. The failure mechanisms we focus on are electromigration (EM), time dependent dielectric breakdown (TDDB) and thermal cycling (TC), which are commonly referred failure mechanisms ([19], [12]).

Electromigration leads to hard failures such as opens and shorts in metal lines, due to migration of atoms in the interconnect lattice. EM is well studied in literature (e.g. [9]) and the failure rate based on Black's model is given in equation (1).

$$\lambda_{EM} = A_0{'}(J - J_{crit})^{-n} e^{(-E_a/kT)} = \lambda^{EM} e^{(-E_a/kT)} \qquad (1)$$

Time dependent dielectric breakdown is a wear out mechanism of the gate dielectric, and it is caused by the electric field and temperature. Failure rate caused by TDDB in the field driven model ([3], [12] is given in equation (2).

To calculate EM and TDDB, average values derived from measurements on a test chip at 95nm technology are assumed for the material dependent and electrical parameters. The equations used in our simulations are provided in the last parts of (1) and (2), where λ^{EM} and λ^{TDDB} are the average values for EM and TDDB measurements respectively.

$$\lambda_{TDDB} = A_0{'} e^{\gamma E_{ox}} e^{(-E_a/kT)} = \lambda^{TDDB} e^{(-E_a/kT)} \qquad (2)$$

Thermal cycling is caused by the large difference in thermal expansion coefficients of metallic and dielectric materials, and it can lead to creating of cracks and other permanent failures. The thermal cycling effect is modeled by the Coffin-Mason equation [9]. Slow thermal cycles happen because of low frequency power changes such as power on/off cycles of a system during a day. Fast cycles occur in much higher frequencies due to events such as power management decisions. Fast cycles gain importance as power management gets more aggressive; so, we focus on modeling fast cycles in this work. The equation for calculating the number of cycles to a thermal cycling induced failure is given in [9], and the corresponding failure rate equation is presented equation (3). C_o is a material dependent constant and q, C_1 and C_2 are empirically determined constants. For brittle structure, commonly used values for q are between 6-9 [12].

$$\lambda_{TC} = C_o{'}[C_1(T_{max} - T_{min}) - C_2(T_{avg} - T_{mold})]^q f_s \qquad (3)$$

Thermal cycling depends on the temperature range ($\Delta T = T_{max} - T_{min}$), frequency of cycles ($f_s$), the average temperature T_{avg} and the molding temperature. For slow cycles, the frequency of cycles and the temperature range is calculated over the lifetime of the system. However, for fast thermal cycles, phases with frequent switching between high and low temperature (i.e. active and sleep states), and phases with stable temperature cannot be distinguished if computation is performed over the lifetime of system. Therefore, we use a *sliding window* to collect the temperature and frequency data for the TC computation. Figure 1 presents the failure rates for three cases: i)No thermal cycling effect taken into consideration; ii)ΔT, T_{avg} and f_s are calculated over the lifetime of the system; iii)ΔT, T_{avg} and f_s are calculated using a sliding history window. The variations in the frequency of thermal cycles can be observed in series (iii), whereas calculating the TC effect over the core's lifetime (ii) does not allow for this observation.

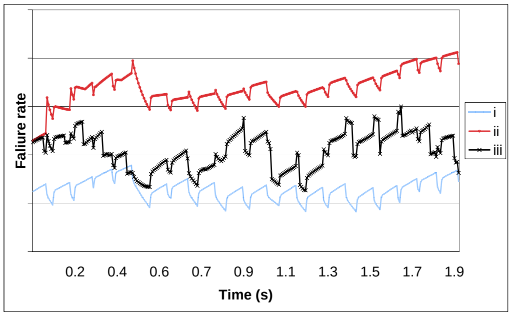

Figure 1: Effect of using a sliding window for TC computation

System failure rate is calculated considering the functional relation among cores and the failure rate per each core. We compute the core failure rate through the sum-of-failure-rates model (SOFR), in which the different failure rates are assumed to be independent and the core failure rate is calculated as the sum of all individual rates. The failure rate of the system is calculated based on the topology of components, considering whether there are redundant processing units [14].

4. RESULTS

In this section, we provide several example cases of reliability analysis performed with the proposed simulation methodology. We use the simulation framework to analyze reliability effects of design choices such as thermal packaging selections and core placement on chip, and effects of runtime events such as power management policies and workload distribution among the cores. We provide failure rate plots and mean-time-to-failure (MTTF) computations for cores.

MTTF is calculated based on the average failure rate over the whole execution time, which is 4 minutes in all of the simulations.

The example SoC we simulate in this work consists of four homogenous cores. The cores are placed on the die symmetrically forming a square. The power and frequency data (provided in Table 1) we use for each core are based on Intel XScale [21]. We assume typical core sizes available in deep submicron process technologies.

State	Active(mW)	Idle(mW)	Freq(MHz)
P1	925	260	624
P2	747	222	520
P3	279	129	208
P4	116	64	104
Psleep	0.163	0.163	0

Table 1: XScale Power Data

The HotSpot input parameters for the baseline medium-expensive package used in our simulations are taken from [18]. For initializing the heat sink, we use the steady state temperature values. We modify dimensions and properties of the thermal package for analyzing packaging effects on reliability, as explained in Section 4.2.

4.1 Reliability Effects of Runtime Events

In this section, we demonstrate how system reliability is affected by the runtime phenomena on chip. We start with analyzing power management policies. When there are longer idle times, DPM achieves much higher power savings than DVS, as the cores spend a considerable time in the sleep state. On the other hand, DVS has less performance overhead because of the negligible transition time spent during frequency switching. Figure 2 provides the average MTTF of the four cores in the SOC for systems under no power management, DPM, DVS and DVS with sleep. DVS with sleep is a combination of the two policies described previously. We provide results for three task sets of 20, 30 and 50 tasks, which result in 26%, 40% and 65% system utilization respectively in the baseline case without power management.

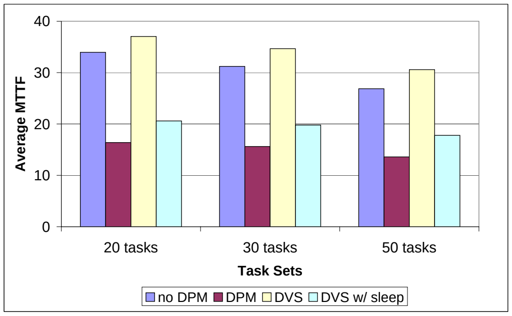

Figure 2: MTTF - comparing power management policies

For the 30-task set, DPM achieves 30% power savings while DVS savings stays below 25%. DVS with sleep reaches more than 40% of savings. The fall in MTTF with DPM is caused mainly by the thermal cycling impact. DVS manages to keep a more stable temperature overall and reduces the thermal cycling problem dramatically. These results show that, to achieve high power savings and high reliability at the same time, combination of DPM and DVS strategies is beneficial. As the number of tasks are increased, the temperature on chip gets higher and decreases the MTTF. This decrease is less noticable when DPM is applied, because higher utilization reduces the frequency of thermal cycles.

Horizontal heat flow plays a significant role in determining the temperature on each unit. Here we look at different workload scenarios on SoCs that are managed by DPM. Again we simulate the four-core SoC described previously. In the first workload scenario, *batch*, the system receives a batch of tasks to execute, and then stays idle for a while. This is a typical scenario in parallel processing systems where the workload arrives in bursts. In the second workload scenario, *producer-consumer*, we assume data dependence among the cores. Two of the cores wait for the others to finish their computation in order to start executing the tasks. In this case, while two cores are active, the other two are in sleep state, and vice versa. This workload distribution is typical in systems that have producer-consumer relation, such as systems in multimedia domain. We simulate these workload schemes with a medium system utilization of 54% in both experiments. When DPM is applied to *batch*, the average system MTTF decreases from 34 years to 24 years. *Producer-consumer* scenario increases the MTTF per core by 23% in average with respect to the *batch* workload.

Figure 3 shows the failure rates of the two workload scenarios. We show results on only one of the cores in the *batch* system as all cores exhibit very similar temperature and failure rate curves. In the other system, we present results on one of the producer-consumer (PC) pairs. The PC cores keep the failure rate lower and more stable due to the heat sharing. If the workload is balanced to maximize heat sharing among cores in SoCs, the reliability impact of DPM can be minimized. This figure is also an example of how failure rate can be monitored at runtime with the proposed methodology.

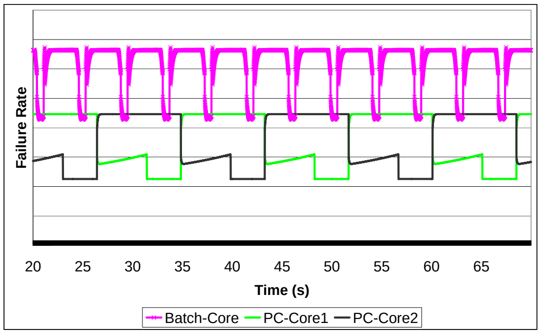

Figure 3: Comparing failure rates in Batch and Producer-Consumer(PC) workloads.

4.2 Reliability Effects of Design Choices

In this section we look into how the location of hot spots on chip effect reliability. Location of hot spots can vary with placement as well as the workload distribution. We compare two systems: i) Two hot cores are adjacent to each other, and the rest of the chip contains relatively cooler units; ii) Two hot cores have a cold structure in between. Figure 4 shows the two systems.

We simulate both systems without applying power man-

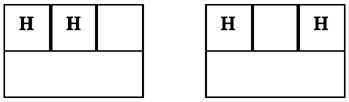

Figure 4: Comparing core placements on chip, H shows the hot cores

agement, and we highly load the two cores to increase their temperature. The peak temperature on the core decreases when some of the heat is transfered to cooler neighbor cores. The MTTF for one of the hot cores in the first system is 30.78 years. In the second placement, the MTTF of the same core is computed as 33.26. Increasing the spacing between the hot spots increases the system MTTF by 8%. The benefits of thermal aware placement become more significant for larger systems.

Depending on the size, cost and application of the systems, a variety of packages are designed in the industry. Thermal packages are generally characterized by the *junction-to-ambient thermal resistance (Θ)*, which is the overall thermal resistance between the die and the surrounding air. The package model we investigate contains die, heat spreader, interface material and heat sink [18]. Based on the thermal resistance data provided by the industry(e.g.[5], [13]), we modify the dimensions and properties of the thermal package in the thermal simulator. Figure 5 provides the average MTTF results comparing a sophisticated package, a medium-expensive package and a very simple package without heat sink. Cheap packages without heat sinks are used in lightweight portable systems where cost and area are hard constraints. These packages, enhanced, normal and without heatsink, have Θ of 33, 45 and 76 Celsius/Watt respectively. We perform the simulations using the task set of 30 tasks on the homogenous four-core SoC. Systems with low cost packages suffer considerably from temperature induced failure rates, as shown in the figure. The lack of heat sink causes higher temperatures as well as higher frequency and magnitude of temperature cycles. Design and runtime methods such as thermal aware placement and workload balancing are crucial for systems without sufficient cooling solutions.

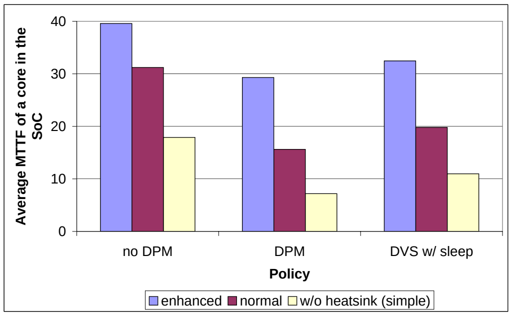

Figure 5: Comparing average MTTF for different thermal packages.

5. CONCLUSION

In this paper we have introduced a simulation methodology for analyzing reliability of multi-core SoCs. The simulator is used for evaluating the reliability of SoCs, considering design and runtime phenomena that affects temperature induced hard failure mechanisms. Power management policies, workload distribution, thermal packaging, and core placement on chip are studied to observe their influences on reliability. Our methodology enables detecting and correcting issues that may arise in design of reliable and power managed SoCs.

When power management causes large temperature variations on chip, it can increase the overall system failure rate due to thermal cycling. Our results show that with workload balancing and DPM/DVS implementations that match system characteristics, we can significantly improve system reliability. The selection of packages for reliable MP-SoCs is a crucial aspect of system design. We demonstrate that for low-cost packages, temperature differentials can be significant enough to cause more frequent system failures.

6. REFERENCES

[1] L. Benini, D. Bertozzi, A. Bogliolo, F. Menichelli, and M. Olivieri. Mparm: Exploring the multi-processor soc design space with systemc. *Springer J. of VLSI Signal Processing*, 41, no. 2, 2005.

[2] L. Benini, A. Bogliolo, and G. D. Micheli. A survey of design techniques for system-level dynamic power management. *IEEE Transactions on VLSI*, 8, No. 3:299–316, 2000.

[3] R. Degraeve and J. O. et. al. A new model for the field dependence of intristic and extrinsic time-dependent dielectric breakdown. *IEEE Transactions on Elect. Devices*, 45, n.2, Feb 1998.

[4] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy. Compact thermal modeling for temperature-aware design. In *DAC*, 2004.

[5] Intersil. Thermal design considerations, application note 1096. *www.intersil.com.*

[6] A. Karlin, M. Manesse, L. McGeoch, and S. Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, pages 542–571, 1994.

[7] W. Kim, D. Shin, H.-S. Yun, J. Kim, and S. L. Min. Performance comparison of dynamic voltage scaling algorithms for hard real-time systems. In *IEEE RTAS*, 2002.

[8] C. Locke, D. Vogel, and T. Mesler. Building a predictable avionics platform in ada: a case study. In *IEEE Real-Time Systems Symposium*, 1991.

[9] H. Nguyen. Multilevel interconnect reliability on the effects of electro-thremomechanical stresses. *Ph.D. dissertation, Univ. of Twente, Netherlands*, March 2004.

[10] P. Pillai and K. G. Shin. Real-time dynamic voltage scaling for low-power embedded operating systems. In *SOSP*, pages 89–102, 2001.

[11] P. Rong and M. Pedram. Power-aware scheduling and dynamic voltage setting for tasks running on a hard real-time system. In *ASPDAC*, 2006.

[12] Semiconductor device reliability failure models. *International Sematech Technology Transfer document 00053955A-XFR*, 2000.

[13] F. Semiconductor. Heatsink small outline package (hsop), application note 2388. *www.freescale.com.*

[14] T. Simunic, K. Mihic, and G. D. Micheli. Reliability and power management of integrated systems. In *DSD*, 2004.

[15] T. Simunic, K. Mihic, and G. D. Micheli. Optmization of reliability and power consumption in systems on a chip. In *PATMOS*, 2005.

[16] K. Skadron et al. Control-theoretic techniques and thermal-rc modeling for accurate and localized dynamic thermal management. In *HPCA*, 2002.

[17] K. Skadron. Hybrid architectural dynamic thermal management. In *DATE*, 2004.

[18] K. Skadron, M. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *ISCA*, 2003.

[19] J. Srinivasan, S. Adve, P. Bose, J. Rivers, and C. Hu. Ramp: A model for reliability aware microprocessor design. *IBM Research Report*, 2003.

[20] J. Srinivasan and S. V. Adve. Predictive dynamic thermal management for multimedia applications. In *ICS*, 2003.

[21] Intel pxa270 processor electrical, mechanical and thermal specification data sheet. *www.intel.com.*