Journal of
Low Power Electronics
Vol. 2, 56–69, 2006

Analysis and Optimization of MPSoC Reliability

Ayse Kivilcim Coskun,[1] Tajana Simunic Rosing,[1,*] Kresimir Mihic,[2]
Giovanni De Micheli,[3] and Yusuf Leblebici[3]

[1] *CSE Department, University of California San Diego, La Jolla, CA 92093, USA*
[2] *Magma Design Automation, Santa Clara, CA 95054, USA*
[3] *Ecole Polythecnique Federale de Lausanne, Lausanne 1015, CH, Switzerland*

(Received: 16 December 2005; Accepted: 23 January 2006)

Advancements in technology enable integration of multiple devices on a single core, resulting in increased on chip power and temperature densities. Higher temperatures, in turn, present a significant challenge for reliability. In this work we propose a comprehensive framework for analyzing reliability of multi-core systems, considering permanent faults. We show that aggressive power management can have an impact on reliability due to temperature cycling. Our cycle-accurate simulation methodology shows fine-grained variations of device failure rates over short time scales, thus enabling workload analysis and scheduling to control the reliability impact. On the other hand, the statistical reliability simulator and optimizer give a view into the long time horizon reliability analysis—over system lifetime, and help us optimize a power management policy under reliability and performance constraints. We show that our optimization strategy can achieve large power savings while still meeting the reliability and performance constraints.

Keywords: Reliability, Power Management, Optimization Methods, Simulation Methodologies.

1. INTRODUCTION

Systems on a chip have an increasingly larger number of cores, communication, and storage elements integrated. Such a large density of transistors integrated on a single chip leads to high power and temperature, and therefore raises significant issues for both power management and reliability.

Several techniques have been proposed to manage power consumption, such as *dynamic voltage scaling* (DVS) and *dynamic power management* (DPM). DVS reduces the power consumption by scaling down voltage and frequency of processor(s) during active periods.[25] DPM, on the other hand, saves power by shutting down system components when they are idle.[27]

Reducing the overall power dissipation by DPM or DVS lowers the average device temperature, and therefore also decreases the rate of temperature-driven hard failures.[14] On the other hand, aggressive power management policies can decrease the overall component reliability because of the degradation effect that temperature cycles have on IC materials.[6, 14] Figure 1 shows a tradeoff between mean-time-to-failure (MTTF) and power savings for three common failure mechanisms: *Electromigration* (EM), *time dependent dielectric breakdown* (TDDB), and *thermal cycling* (TC). In our work we use measured values of these failure rates on a test core obtained from a silicon manufacturer for 95 nm technology. This particular test core has a single sleep state and no DVS capability. As shown in the figure, when power management becomes more aggressive, it helps to improve MTTF due to electromigration (EM) and time dependent dielectric breakdown (TDDB). Nevertheless, it results in a significant cost in terms of the reliability impact of thermal cycles, due to the large temperature differentials and frequent shutdowns caused by the policy. TC effect dominates the overall MTTF as power savings increase.

In this work we provide a comprehensive modeling and optimization methodology for analyzing reliability of systems on a chip. We propose two levels of reliability modeling: statistical and cycle accurate. Analyzing reliability with both approaches enables us to observe long-term trade-offs between power management and reliability, as well as short-term changes on the system failure rate when power management or scheduling policies vary.

In the statistical model, we evaluate long term reliability, performance, and power consumption of cores in SoCs by computing system reliability as a function of failure rates, system configuration, and management policies. With our statistical model we define and solve a joint dynamic

*Author to whom correspondence should be addressed.
Email: tajana@cs.ucsd.edu

 1546-1998/2006/2/056/014 doi:10.1166/jolpe.2006.007

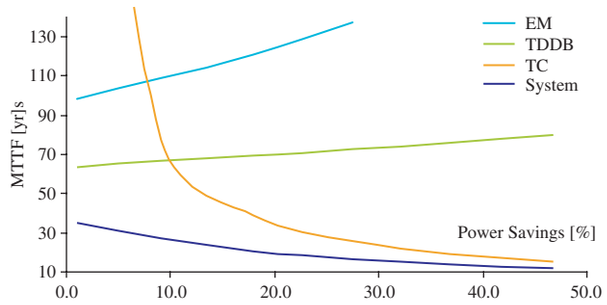

Fig. 1. Failure mechanisms in 95 nm technology for a single core.

power management (DPM) and reliability (DRM) optimization problem. Experimental results show that with our joint optimization method we can save energy by 40% while meeting both reliability and performance constraints. While statistical modeling can provide analysis on long-term reliability of systems, in order to model effects of policy changes, and workload scheduling precisely, workload simulation at cycle accurate level is needed. We integrate cycle-accurate workload modeling with power management, temperature modeling, and failure rate modeling to obtain fine-grained analysis of system reliability. Our simulator analyzes failure rate variations of multi-core systems considering several power management policies, workload distribution schemes, and system topologies.

The rest of the paper begins with an overview of related work. Temperature and reliability models are explained in Section 3. Section 4 provides details on the statistical and cycle-accurate simulators. Optimization methodologies are described in Section 5. We present the experimental results in Section 6 and Section 7 concludes.

2. RELATED WORK

Power management in systems on chips has been extensively discussed in literature. Two commonly referred techniques are dynamic power management (DPM) and dynamic voltage scaling (DVS). The capability of these techniques to perform accurate estimations determines the efficiency of the policy, especially in the case of non-stationary workloads where the frequency of workload arrivals varies over time. Chung et al. introduced online workload learning techniques based on sliding observation windows in order to estimate future workload accurately and apply DPM.[27] A workload decomposition technique is proposed in Ref. [28] to predict the on-chip activity for performing DVS with higher power savings. For large SoCs, a stochastic optimization methodology based on a closed-loop control model for a unified dynamic voltage and power management at core-level has been presented in Ref. [9]. This optimization includes both node and network-centric views of systems. Benini et al. provides a detailed summary of system-level power management techniques in Ref. [26].

Although power management typically reduces the core temperature, it is not always capable of eliminating high temperature spikes. Thus, thermal modeling and dynamic thermal management (DTM) have become important. HotSpot[2] is a well-known example of a dynamic architecture-level thermal simulator. Recent approaches for dynamic thermal management (DTM) target architecture level with the goal of reducing power density in thermal hot spots by employing a number of different techniques such as global clock gating,[1] migrating computation to spare units,[2] traffic rerouting,[3] predictive techniques for multimedia applications,[4] and DVS specifically for thermal management.[2–4] DTM's goal is to eliminate thermal hot spots at run time, but it does not directly consider long-term reliability.

Reliability of SoCs has become of increasing concern due to the high permanent and transient faults in deep submicron and nanoscale technologies. In this work we focus on temperature-induced hard failure mechanisms. Typical examples of hard failures include open interconnect lines, shortening between adjacent metal layers and crack formations; a good overview can be found in Ref. [13]. For example, electromigration failure mechanism caused by temperature gradients is investigated in Ref. [14]. The description of the connection between fast thermal cycling and thin film cracking is presented in Ref. [15] and a failure rate model is given in Ref. [14]. Time-Dependent Dielectric Breakdown (TDDB) has also been studied extensively, and a model for TDDB is discussed in Ref. [16].

Improving system reliability and increasing processor lifetime by implementing redundancy at the architecture level are discussed in Refs. [5, 8, 29]. The RAMP simulator models microarchitecture MTTF as a function of temperature related failure rates of individual structures on chip due to intrinsic hard failure mechanisms.[11] RAMP gets activity estimates from performance simulation with an instruction level simulator and uses them to obtain power consumption and temperature of microarchitecture components. Reliability computation is performed using the temperature values. Exploiting the architecture's knowledge of the runtime workload distribution, the DTM methodology proposed keeps processor temperature under a given threshold by architectural adaptation and applying dynamic voltage scaling.

In our work we study both the long-term system reliability as a function of power management policies as well as the short-term variations in system and core reliability. Our simulation is targeted at evaluating behavior of large, multi-processor SoCs, with workloads normally experienced during their useful life. We consider electromigration, time dependent dielectric breakdown, and fast thermal cycling as the sources of hard failure mechanisms. Our statistical methodology presents for the first time a unified approach for the joint optimization of reliability, power consumption, and performance in SoCs. In addition

to long time horizon statistical modeling and simulation of SoC reliability, we also correlate the real workload on chip and power management policy changes with the short-term reliability effect. Our fine-grained model shows a clear need for careful dynamic workload scheduling in order to improve the overall system reliability. In order to be able to study these effects, we next discuss how we have enhanced the traditional statistical and cycle accurate power and performance simulators with thermal and reliability models.

3. TEMPERATURE AND RELIABILITY MODELING

3.1. Thermal Modeling

Detailed thermal modeling has received significant attention recently due to skyrocketing temperatures of current systems. The well-known analogy between thermal phenomena and an RC circuit models temperature gradient at each node and heat flow among structures using thermal capacitance and resistances respectively. We use Hotspot version 2 (Ref. [2]) to provide temperature samples based on the cycle-accurate power and performance simulations. HotSpot computes the vertical and horizontal thermal resistances and capacitances, given the dimensions and material properties of units and chip. It includes a very detailed thermal package model for a typical package set-up for today's chips. The simulator takes into account the lateral thermal diffusion on chip as well, which increases its accuracy. In our simulations, we set the convection resistance and capacitance values from the chip to air as those of a medium expensive package, and assume typical core sizes available in deep submicron process technology.

Our statistical simulation platform uses a simplified model because of very long simulation time horizons. The temperature in a state is estimated using the base active state temperature, the time spent in a state (e.g., *active, idle, sleep*) due to an action and the state's steady state temperature. The active state temperature is defined using the thermal resistances of die and package, R_{thdie} and $R_{\text{thpackage}}$, for a reference frequency and voltage of operation in the active state. Thermal RC constant, $\tau \approx c\rho a^2$, has $c = 10^6$ J/m^3 K for silicon thermal capacitance and $\rho = 10^{-2}$ mK/W for thermal resistivity.[12]

3.2. Failure Rate Modeling

We use the temperature estimates obtained from thermal models to calculate the reliability of components during their useful life. In the statistical model, the failure rates are considered constant within any given operational state (i.e., active, idle, and sleep). In the cycle-accurate model, our goal is to observe the variations in the failure rate, so we compute the failure rate at each sampling interval. We consider three failure mechanisms commonly referred in the literature: *Electromigration* (EM), *Time Dependant Dielectric Breakdown* (TDDB), and *Thermal Cycles* (TC).

Electromigration is the transfer of material resulting from the gradual movement of ions in the conducting path due to the momentum transfer between conducting electrons and metal atoms. It leads to permanent faults such as opening of metal lines/contacts, shortening between adjacent metal lines, etc. The MTTF due EM process is commonly described by Black's model:

$$MTTF_{\text{EM}} = A_{\text{o}}(J - J_{\text{crit}})^{-n} e^{(E_{\text{a}}/kT)} \tag{1}$$

where A_{o} is an empirically determined constant, J is the current density in the interconnect, J_{crit} is the threshold current density, and k is the Boltzmann's constant, $8.62 \cdot 10^{-5}$. For aluminum alloys E_{a} and n are 0.7 and 2 respectively. EM failure rate is modeled for idle and active states only, because leakage current present in the sleep state is not large enough to cause the migration. We formulate the EM failure rate as a product between an average measured value in a given power state s, $\lambda^{\text{EM}}_{m,s}$, and a factor that is a function of temperature.

$$\lambda^{\text{EM}}_{\text{core},\,s} = A'_{\text{o}}(J_s - J_{\text{crit}})^n e^{(-E_{\text{a}}/kT_s)} = \lambda^{\text{EM}}_{m,s} e^{(-E_{\text{a}}/kT_s)}; \quad \forall\, s = \textit{active}, \textit{idle} \tag{2}$$

Time Dependent Dielectric Breakdown is a wear out mechanism of dielectric due to electric field and temperature. The mechanism causes the formation of conductive paths through dielectrics shortening the anode and cathode. MTTF due to TDDB can be defined with the field-driven model:

$$MTTF_{\text{TDDB}} = A_{\text{o}} e^{-\gamma E_{\text{ox}}} e^{(E_{\text{a}}/kT)} \tag{3}$$

where A_{o} is an empirically determined constant, γ is the field acceleration parameter, and E_{ox} is the electric field across the dielectric. The activation energy, E_{a}, for intrinsic failures in SiO_2 is found to be 0.6–0.9 and for extrinsic failures about 0.3.[13] The failure rate due to TDDB mechanism for active, idle, and sleep state is defined as a product between an average measured value in a given power state s, $\lambda^{\text{TDDB}}_{m,s}$, and a factor that is a function of temperature we obtain from the thermal models.

$$\lambda^{\text{TDDB}}_{\text{core},\,s} = A'_{\text{o}} e^{\gamma E_{\text{ox},s}} e^{(-E_{\text{a}}/kT)} = \lambda^{\text{TDDB}}_{m,s} e^{(-E_{\text{a}}/kT)}; \quad \forall\, s = \textit{active}, \textit{idle}, \textit{sleep} \tag{4}$$

Temperature Cycling effect is caused by the large difference in thermal expansion coefficients between metallic and dielectric materials, the silicon substrate and the package. It can cause plastic deformations that eventually create cracks, fractures, short circuits, and other related failures. The effect of low frequency thermal cycles (processor on/off) has been well studied by packaging community and can be modeled by Coffin-Manson model.[13] Thermal cycles that occur with higher frequencies are gaining importance as features sizes get smaller and low-k dialectic is introduced to the fabrication process. Recent

work[14] shows that such cycles play a major role in cracking of thin film metallic interconnects and dielectrics. Expected number of fast thermal cycles before core failure is given in Eq. (5). It does not only depend on the temperature range ($T_{\max} - T_{\min}$) but is also strongly influenced by the average temperature, T_{avg} and the molding temperature of the package process, T_{mold}. The exponent q ranges from 6–9, and $C_{1,2}$ are fitting constants defined in Ref. [14] for on chip structures.

$$N_f = C_o[C_1(T_{\max} - T_{\min}) - C_2(T_{\text{avg},\,s} - T_{\text{mold}})]^{-q} \quad (5)$$

For a power-managed core, two distinct thermal cycle loops exist. The first one is between active and idle states and has a relatively small temperature difference so is unlikely to experience large thermal cycles. On the other hand, the transitions in and out of sleep states can have a large difference in power consumption, and occur infrequently enough to cause the temperature to vary significantly. Therefore, we can calculate the total failure rate due to the thermal cycling effect as shown in Eq. (6). T_{avg} and T_{mold} are the average temperature and the temperature of package molding process, while Cs are fitting constants. $T_{\min}$ and $T_{\max}$ are the minimum and maximum temperatures observed and f_s is the frequency of transitioning into the sleep state.

$$\lambda^{\text{TC}}_{\text{core},s} = C_o'[C_1(T_{\max} - T_{\min}) - C_2(T_{\text{avg}} - T_{\text{mold}})]^q$$
$$f_s = \forall\, s = sleep \quad (6)$$

The temperature values are obtained differently for the statistical and cycle-accurate model. In statistical model $T_{\max}$ and $T_{\min}$ correspond to T_{active} and T_{sleep} respectively and T_{avg} and f_s are calculated over the lifetime of the system. In contrast to the life-time analysis, cycle-accurate simulations model temperature, and failure rate at each cycle. So, the $T_{\max}$, $T_{\min,}$ T_{avg}, and f_s values are calculated by using a sliding window over a last (fixed) set of temperatures and transitions. Utilizing the sliding window enables to observe the varying frequency of sleeping and to distinguish between consecutive phases of rather stable temperature and phases with frequent state switching.

In order to calculate the overall component failure rate, we assume that the individual failure rates are statistically independent and thus we can use a *sum-of-failure-rates* (SOFR) model. A core's failure rate is represented as the sum of the failure rates of individual failure mechanisms.

3.3. System Reliability

In statistical modeling and simulations, since we are interested in assessing the reliability over the typical operation time, we assume that the failure rates are constant in time. Cycle-accurate modeling tracks the variations in failure rate over time. During the useful life of devices the failure rate is usually modeled with the exponential distribution. Thus we can represent the component reliability with a failure rate, λ_f as follows: $R(t) = e^{-\lambda_f t}$, with mean time to failure $MTTF = 1/\lambda_f$. Since a large SoC system has multiple cores that can act as "back-up" elements to each other, we take the system topology into account for our reliability calculations. In a system, components are in *series (parallel)* if the overall correct operation hinges upon the *conjunction (disjunction)* of the correct operation of components. The overall system reliability can be calculated by applying the rules for series and parallel composition, under the assumption that failure rates are statistically independent.

$$R_{\text{system}}(t) = \prod_{i-0}^{n} R_i(t) \Rightarrow R_{\text{system}}(t) = e^{-\sum_{i=0}^{n} \lambda_{f_i} t} \quad (7)$$

The system built with n series components fails if any of its components fails. When failure rates are constant, the failure rate of a series composition is the sum of the failure rates of each component as shown in Eq. (7). Alternatively, the parallel combination fails only if all n components those are in parallel fail:

$$R_{\text{system}}(t) = 1 - \prod_{i-0}^{n}(1 - R_i(t)) \quad (8)$$

Systems with parallel structures have built-in redundancy. Such systems can either have all components concurrently operating (*active parallel*) or only one component active while the rest are in low power mode (*standby parallel*). Active parallel combination has higher power consumption and lower reliability than standby parallel, but also faster response time to failure of any one component. The combined failure rate of M active components, λ_{fap}, is defined using binomial coefficient, C_i^M, and active reliability rate, λ_f as shown in equation below.

$$\lambda_{\text{fap}} = \sum_{i=1}^{M}(-1)^{i-1}\frac{C_i^M}{i\lambda_f} \quad (9)$$

Both power consumption and reliability can be improved if only one core is active at a time and others are in standby, with the cost of performance. The failure rate of M parallel components that are in standby is $\lambda_{fsp} = \lambda_{fs}/M$.[17] To get the overall failure rate we need to combine M standby components with one active parallel component.

4. SIMULATION

We introduce two simulation frameworks in this paper: cycle-accurate and statistical. Cycle-accurate reliability simulation provides fine grained analysis of the effect power management and workload scheduling have on the system reliability. The statistical simulation platform is used for computing long-term system reliability with a state-machine model. We next outline the both methodologies.

4.1. Cycle-Accurate Reliability Simulation

Our cycle-accurate simulation methodology is the first to enable evaluation of fine grained changes in the reliability of multi-core SoCs due to temperature induced failure mechanisms. The simulation framework consists of the following major parts:

(1) Cycle accurate multi-core power simulator;

(2) Power Manager, responsible for applying power management and dynamic voltage scaling policies;

(3) Thermal modeling tool that calculates the instant temperature at each simulation step;

(4) Failure rate modeling tool that outputs failure rates with respect to temperature and SoC characteristics. Figure 3 presents an overview of the simulator.

A cycle-accurate multi-core power simulator models core interactions in order to obtain the workload traces for each core. We use MP-ARM,[32] a multi ARM core simulator, to obtain each core's utilization trace. Since in a multi-core scenario it takes a very long time to simulate real traces; we chose computational kernels as representative samples. The benchmarks we simulate perform matrix computations on a SoC model, which are typical operations in multimedia applications. The cores interact with each other through the shared memory in a producer–consumer relationship, implementing a pipeline of operations. This represents a typical set-up for communication among cores on chip in the lack of OS support. Two computation kernels are selected for simulations in this paper; one with high utilization and other with lower utilization. In many applications, for example multimedia, certain parts of the workload either repeat frequently, or exhibit similar statistics.[31] Therefore, we replicate each sample trace to analyze longer-term execution phases.

The second stage of the simulator is the *Power Manager*, which applies the DPM and DVS strategies on the core utilization trace it takes as input. For simplicity in implementation, we used a Markov policy.[10] The Markov policy provides an average efficiency of the power manager over all workloads.

DVS policies typically make a decision on the next frequency/voltage setting based on the utilization ratio in the observation window. Based on the given workload trace, our power manager selects the best DVS policy that could be applied for a given set of frequency/voltage pairs available in the core. The power manager for DVS is configured to behave as an oracle with perfect knowledge of future workload in order to emphasize the differences between DPM and DVS.

Thermal modeling is performed by HotSpot[2] as discussed Section 3. The last stage of the cycle-accurate simulator is *failure rate modeling*. The failure rates are computed using the equations given in Section 3 and the temperature trace obtained from thermal modeling tool. A significant differentiation between the statistical and cycle-accurate models is that the cycle accurate model calculates temperature values and failure rates of each core at every sampling interval, rather than estimating an average over a long period of time. This way we can observe the detailed failure rate variations resulting from workload and policy changes. The modeling of the short-term changes in these rates enables applying adaptive reliability policies in real-life systems.

4.2. Statistical Simulation Methodology for SoC Reliability

The statistical reliability simulator is the first one to unify high level modeling of voltage scaling, power management, and reliability. Previous work has introduced Microarchitecture-level[11] or lower level models.[3] The statistical simulator enables analysis of trade offs between power performance and reliability for large multi-core systems over the system's useful life, which is in terms of years. Each core is modeled as a *power and reliability state machine* (PRSM) as shown in Figure 2. Transitions

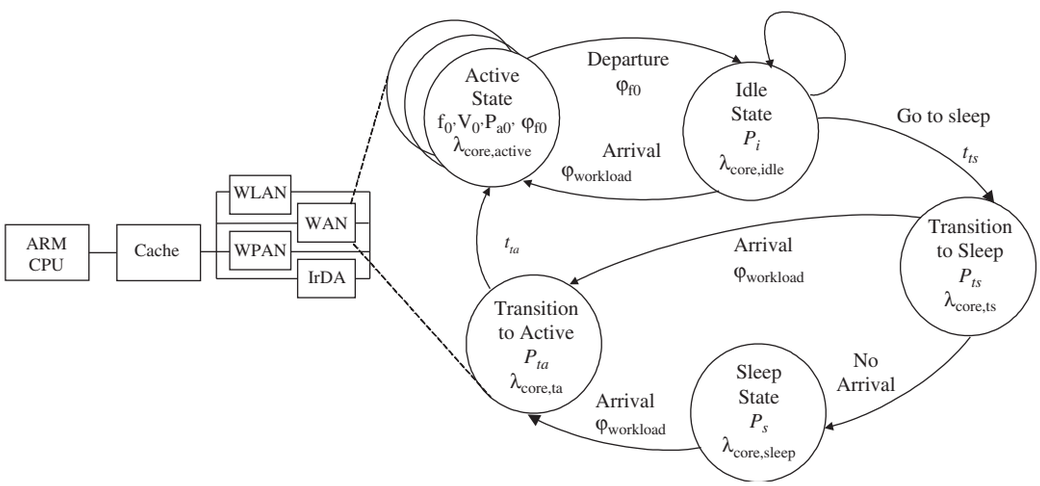

Fig. 2. System model.

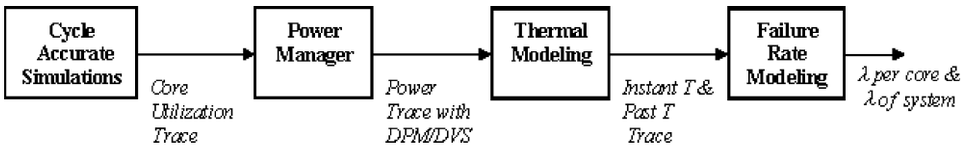

Fig. 3. Cycle-accurate simulation flow.

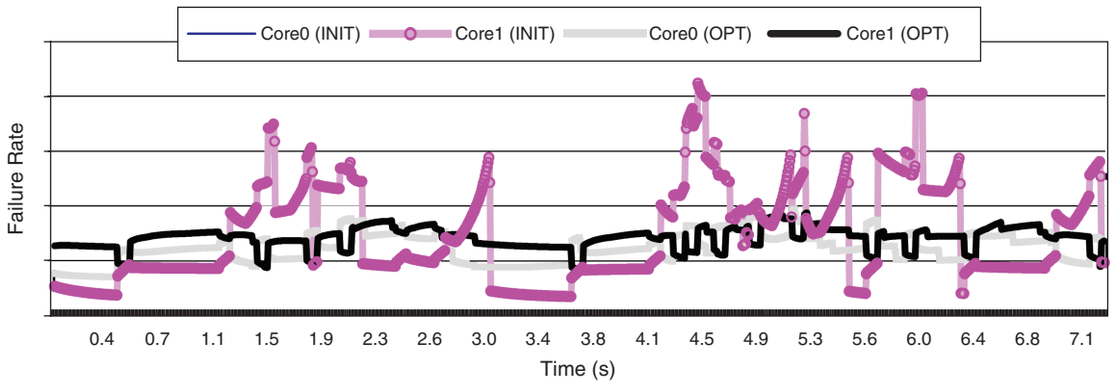

Fig. 4. Workload scheduling to achieve a lower failure rate.

between states occur due to a DVS policy (between different active states), DPM policy (between idle and sleep states) or because of natural core operation, such as arrival of a workload request (arrival arcs) finishing processing on data (departure arcs) or a failure of the core. Each core's failure rates are calculated from the data collected at runtime (e.g. amount of time spent in a state, frequency of state changes) and core's specifications (e.g. power, voltage, transition times) using equations given in Section 3. The simulator identifies and activates spares when one or more cores fail based on the reliability network configuration. Further details are provided in Ref. [30].

We next discuss how workload scheduling can help on more fine-grained level, and optimization for longer time horizons.

5. POWER AND RELIABILITY OPTIMIZATION

5.1. Workload Scheduling for Higher Reliability

Temperature related reliability problems on chip arise due to two major thermal phenomena: high instantaneous temperatures (hot spots) and thermal cycling. Heat flow among structures is very influential in determining the overall system temperature. If a hot core is able to share its heat with a colder neighbor, temperature at the hot spot is reduced. This heat sharing property can be exploited in order to maintain a lower overall temperature. Many applications finish execution much before their worst-case execution times and thus hardware resources are under-utilized.[25] Moreover, interacting units typically depend on each other for data and stay idle when they are waiting for new workload arrival from the producer. Such idle times can be utilized for optimizing scheduling for higher reliability.

Figure 4 shows a sample scenario. Here we demonstrate the failure rates of each core in a two-core system, obtained from the cycle-accurate simulator. The initial core (INIT) has 2 cores managed by DPM, and the two cores have exactly the same workload distribution (e.g. Fig. 5), thus the same temperature and failure rate. In this case the cores are simultaneously active, which causes large amount of increases in temperature. In the optimized case (OPT), the workload distribution of one of the cores is adjusted such that it is active whenever the other core is idle (or at sleep). Figure 6 shows such a workload

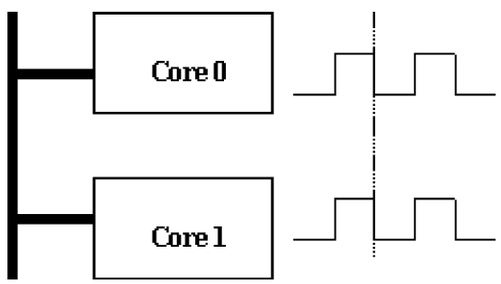

Fig. 5. Cores with identical workload utilization.

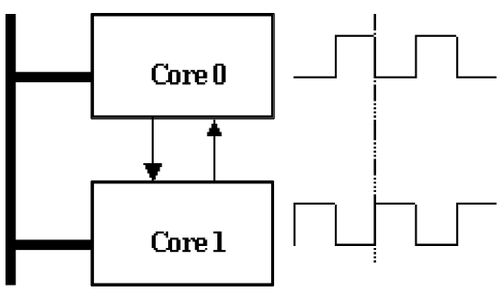

Fig. 6. Cores with non-overlapping active times.

utilization scheme. This workload schedule maximizes the benefit of heat sharing among the cores and dramatically reduces and stabilizes the failure rate.

5.2. Power Management with Reliability Constraints

In this section we define the optimization problem given a system topology and a set of component states (active, idle, etc.) characterized by failure rate, power consumption, and performance. The result of optimization is a power and reliability management policy to be implemented at a system level over the useful life of a device. Policy manager makes decisions at each event occurrence (e.g., arrival or departure of a request). The *interevent time set* is defined as $T = \{t_i\_\text{s.t.}\, i \in 0, 1, 2, \ldots, i_{\max}\}$ where each t_i is the time between two successive event arrivals and $i_{\max}$ is the index of the maximum time horizon. We denote by $s_i \in S_i$ the system state at decision epoch i. Commands are issued whenever the state changes. We denote by $a_i \in A_i$ *an* action (or command) that is issued at decision epoch i. Each core in the reliability network is modeled with a *power and reliability state machine* (PRSM) as shown in Figure 2. PSRM is a state diagram relating service levels to the allowable transitions among them. Multiple cores form a reliability network of series and parallel combinations of single core PRSM models. Single core PRSM characterizes each state by its failure rate, $\lambda_{\text{core, state}}$, and power consumption, P_{state}. Thus, the active state i is characterized by the failure rate $\lambda_{core,\, active\, i}$, frequency and voltage of operation, f_i, V_i, which is equivalent to the core processing rate φ_{fi}, and power consumption P_{ai}. In the active state the workload and core's data processing times follow the exponential distribution with rates $\varphi_{\text{workload}}$ and $\varphi_{\text{core\_}fi}$. In the idle state a core is active but not currently processing data. Sleep state represents one or more low power states a core can enter. *TransitionToSleep* and *TransitionToActive* states model the time and power consumption required to enter and exit each sleep state. Transition times to/from low-power states follow uniform distribution with average transition times t_{ts}, t_{ta}.[26] In the active state, the power manager decides the appropriate frequency and voltage setting, where as in the idle state the primary decision is which low-power state core should transition to and when the transition should occur. The arcs represent transitions between the states with the associated transition times and rates. Table I summarizes all distributions used in modeling performance, power consumption, and failure rates. The choices of distributions have been validated by experimental measurements.[10]

Failure rates change with each power state since different level of power consumption causes a different temperature. We calculate the expected temperature for each state ($E[T_{\text{state}}]$) as a function of the expected time spent in that state, $y(s, a)$, the steady state temperature for the state and the technology parameters ($\tau \approx c\rho a^2$, defined earlier).

Table I. System model characteristics.

State	Distribution	Parameters	Failure Rate
Active	Exponential	$\varphi_{\text{core},\, f-V}$ $\varphi_{\text{workload}}$	$\lambda_a = \lambda_a^{\text{EM}} + \lambda_a^{\text{TDDB}}$
Idle	General	*Collected Data*	$\lambda_i = \lambda_i^{\text{EM}} + \lambda_i^{\text{TDDB}}$
Transition	Uniform	$t_{\min}$, $t_{\max}$	
Sleep	General	*Collected Data*	$\lambda_s = \lambda_s^{\text{TDDB}} + \lambda_s^{\text{TC}}$
Queue = 0			
Queue > 0	Exponential	$\varphi_{\text{workload}}$	

Using the expected temperature we can calculate a stationary value for failure rates due to each mechanism and per each power state.

We can formulate a linear program (LP) for the minimization of energy consumption under reliability constraint as shown in Eq. (10). The first constraint is known as a balance equation since it requires that a number of entries into a given state to equal the number of exists out of that state. The second constraint limits the sum of all probabilities to equal one. Each probability is represented by a product between the expected time spent in a state s under a command a, $y(s, a)$, the unknowns of the LP, $f(s, a)$. The $A \times S$ unknowns in the LP, $f(s, a)$, called *state-action frequencies*, are the expected number of times that the system is in state s and command a is issued. The last three equations specify constraints on performance and reliability. The result of optimization is a globally optimal power management policy that is both stationary and randomized. The policy can be compactly represented by associating a probability of issuing command a when the system is in state s, $x(s, a) = f(s, a)/\sum f(s, a')$ with each state and action pair. We explain now in more detail all variables used in LP formulation starting with the reliability constraint. Both continuous and time indexed versions are presented, labeled with dt and Δt respectively.

$$
\begin{aligned}
\min \quad & \sum_{c=1}^{N} \text{cost}_{\text{energy},\, c} \\
\text{s.t.} \quad & \sum_{a \in A} f(s,a) - \sum_{a \in A} \sum_{s' \in S} m(s'|s,a) f(s',a) = 0; \quad \forall s, \forall c_s \\
& \sum_{a \in A} \sum_{s \in S} y(s,a) f(s,a) = 1; \quad \forall c_s \\
& \sum_{c=1}^{N} \text{cost}_{perf,\, c} < Perf_{\text{const}}; \quad \forall c \\
& Tpl(\lambda_c) \le Rel_{\text{const}}; \quad \forall c_s \\
& \lambda_c = \sum_{i \in F} \sum_{a \in A} \sum_{s \in S} \lambda^i_{\text{core}}(s,a) y(s,a) f(s,a)
\end{aligned}
\tag{10}
$$

The reliability constraint, *Tpl* is a function of the system topology, i.e., *Tpl = f (series, parallel combinations)*. For example, with series combinations $Tpl = \sum \lambda_{\text{core},\, s}$, and with parallel standby $Tpl = \lambda_{\text{core, standby}}/N_{\text{standby}}$. A reliability network may have a number of series and parallel combinations of cores. Each core's failure rate,

λ_c, is a sum of failure mechanisms, $i \in \{\text{EM, TDDB, TC}\}$, when the core is in the state s and the action a is given. For example, the reliability constraint is given in Eq. (11) for a core that has one *active* (A), *idle* (I), and *sleep* (S) state and two actions: *go to sleep* (S) and *continue* (C). Failure rate in each state, $\lambda_{\text{core, state}}$, is a sum of failure rates due to failure mechanisms active for that state as described in Section 3.

$$\lambda_A y(A,C) f(A,C) + \lambda_I y(I,C) f(I,C) + \lambda_I y(I,S) f(I,S) + \lambda_S y(S,C) f(S,C) \leq Rel_{\text{const}} \quad (11)$$

Our model also defines two cost metrics, energy, and performance. The average cost incurred between two successive events as defined in Eq. (12) is a sum of the lump sum cost $k(s_i, a_i)$ incurred when action a_i is chosen in state s_i, in addition to the cost in state s_{i+1} incurred at rate $c(s_{i+1}, s_i, a_i)$ after choosing action a_i in state s_i. The value of cost rate, $c(s_{i+1}, s_i, a_i)$ is power consumption in a state s_i for the calculation of energy cost, and performance penalty for the performance constraint.

$$Cost(s_i, a_i) = \begin{cases} k(s_i, a_i) + \int_0^{\infty} \Big[F(du|s_i, a_i) \\ \quad \times \sum_{s_{i+1} \in S_{i+1}} \int_0^{u} c(s_{i+1}, s_i, a_i) \\ \quad \times p(s_{i+1}|t_i, s_i, a_i)\, dt \Big] \;\; \forall dt \\ k(s_i, a_i) + \sum_{s_{i+1} \in S_{i+1}} c(s_{i+1}, s_i, a_i) y(s_i, a_i) \;\; \forall \Delta t \end{cases} \quad (12)$$

When action a_i is chosen in system state s_i, the probability that the next event will occur by time t_i is defined by the cumulative probability distribution $F(t_i|\, s_i, a_i)$. For example, in the active state the cumulative distribution of request departures is given by $F(active, departure) = 1 - e^{-t\lambda_{\text{core}}}$. The probability that the system transitions to state s_{i+1} at or before the next event t_i is given by $p(s_{i+1}|\, t_i, s_i, a_i)$. For example, in active state the probability of departure is given by $p(idle|t, active, departure) = \lambda_{\text{core}}/(\lambda_{\text{core}} + \lambda_{\text{workload}})$. In time-indexed idle and sleep states the probability of getting an arrival in time increment Δt can be calculated as follows:

$$p(s_{i+1}|t_i, s_i, a_i) = \frac{F(t_i + \Delta t) - F(t_i)}{1 - F(t_i)} \quad (13)$$

The expected time spent in each state s when a command a is given, $y(s, a)$, is defined in Eq. (14). For example, the expected time spent in the active state is equal $1/(\lambda_{\text{core}} + \lambda_{\text{workload}})$.

$$y(s_i, a_i) = \begin{cases} \int_0^{\infty} t \sum_{s_{i+1} \in S_{i+1}} p(s_{i+1}|t_i, s_i, a_i) F(dt|s_i, a_i) & \forall dt \\ \int_{t_i}^{t_i + \Delta t} \dfrac{(1 - F(t))\, dt}{1 - F(t_i)} & \forall \Delta t \end{cases} \quad (14)$$

Finally, the probability of arriving into state s_{i+1} given that the action a_i was taken in state s_i is defined by:

$$m(s_{i+1}|s_i, a_i) = \int_0^{\infty} p(s_{i+1}|t_i, s_i, a_i)\, F(dt|s_i, a_i) \quad (15)$$

For the time indexed states the probability of arriving to the next idle state is defined to be $m(s_{i+1}|s_i, a_i) = 1 - p(s_{i+1}|t_i, s_i, a_i)$ and of transition back into the active state is $m(s_{i+1}|s_i, a_i) = p(s_{i+1}|t_i, s_i, a_i)$. Equations (12)–(15) are sufficient to calculate all variables needed for the LP shown in Eq. (10). The LP can then be solved using any linear program solver in a matter of less than a second. The final output of optimization is a table that specifies probabilities of transitioning cores into each of their low-power states.

6. RESULTS

In this section we discuss the advantages of both modeling frameworks and the benefits of the optimization. We provide detailed analysis on the effects of power management, system topology, and workload scheduling on reliability.

6.1. Cycle-Accurate Simulation Results

We performed cycle-accurate analysis of thermal impact on failure rate considering various workload characteristics, power management policies, and system topologies. The two main traces we use in these simulations are Trace 1 and Trace 2, which have 22% and 70% utilization respectively when run on a single core. These traces represent multimedia kernels with short and long idle times in order to clarify the workload effect on reliability. More details can be found in Section 3. Failure rate variation plots are drawn with respect to the baseline case where no power management is applied. As in the statistical simulations, the parameters in the failure rate equations are derived from measurements in 95 nm process technology. Workload simulations are run on MP-ARM. We use Intel's XScale processor power and performance values for each of the cores in SoC.[7] The performance penalty for the DPM and DVS policies for Traces 1 and 2 are given in Table VI. The penalties are calculated based on the transition time specifications given in the datasheet of XScale processor. The performance penalties given in Table VI below are calculated as $t_{\text{delay}}/t_{\text{total}}$; where t_{delay} is the cumulative delay occurring due to the transition between states or the time spent for changing the frequency settings, and

Table II. SoC parameters.

IP block	P_{active} [W]	P_{idle} [W]	P_{sleep} [W]	t_{ts} [s]	t_{ta} [s]
DSP (TMS6211)[18]	1.1	0.5	0.01	250 u	100 n
Video (SAF7113H)[19]	0.44	0.44	0.07	110 m	0.9
Audio (SST-Melody-DAA)[20]	0.11	0.003	3e−4	6 u	0.13
I/O (MSP43011×2)[21]	1e−3	N/A	6e−6	100 n	6 u
DRAM (Rambus 512M)[22]	1.58	0.37	1e−2	16 n	16 n

Table III. XScale power state characteristics.

State	Active (mW)	Idle (mW)	Freq (MHz)
P_1	925	260	624
P_2	747	222	520
P_3	279	129	208
P_4	116	64	104
P_{sleep}	0.163	0.163	0

t_{total} is the original total execution time of the core. For all of the DPM simulations in this paper we use the same DPM policy as discussed in Section 4.2, which provides roughly 10 to 30 percent of power savings for a single core in the workloads we used.

We assume a 1 mm^2 core, which is a typical size for a core in deep submicron process technologies. The HotSpot input parameters used in our simulations are provided in Table V. For the heat sink initial temperature, we use the steady state values obtained in temperature simulations. The sampling interval in the simulations is 6.2 ms, and the whole simulation time varies between 2 to 8 seconds. This sampling interval provided sufficient precision for observing the power state changes in the simulated benchmarks. We assume each state has a fixed power value (power values given in Table III). This is a reasonable assumption since the core power value does not exhibit significant changes depending on the executed instruction, as it is shown for the StrongArm core in Ref. [10]. We used the same sampling rate of 6.2 ms for the thermal/reliability simulations. HotSpot[2] reported that temperature variations can be observed during time intervals on the order of milliseconds.

Figure 7 demonstrates the failure rate per core for a two-core system with identical core sizes that are placed adjacent to each other. In this system, Core0 is managed with DPM policy and DVS is applied to the other core. Both cores execute a higher utilization trace (#2). The change of failure rate is calculated with respect to the two-core system running the same workload without power management. DVS is able to keep the failure rate more stable than DPM, as DVS managed core is not prone to temperature cycling. The average failure rate of the two cores is very close due to heat sharing between them.

SoCs that are manufactured today contain a number of cores, on-chip memories and other units specialized for management and interfacing. In such large systems, horizontal heat flow among different structures can be very influential on temperature and therefore on reliability. Such systems can benefit from workload scheduling. We next show examples of how scheduling can affect system reliability. In Figure 4, initial core (INIT) consists of two cores processing a lower utilization trace (#1), and both cores are power managed simultaneously with DPM (Fig. 5). The failure rate plots for both of the cores overlap completely. In the optimized system (OPT), Core0 runs Trace 1 whereas Core1 has the opposite utilization of Trace 1, such as in Figure 6. DPM is applied to both cores. Even though the optimized system's utilization is higher than the initial case, failure rate is much lower. The heat sharing among the adjacent cores keeps the failure rate low and stable.

Table IV. Power savings and MTTF increase for XScale.

Policy	Power	MTTF
None	0%	0%
DVS	35%	42%
DPM (Rmax)	16%	6%
DPM (ave)	47%	−12%
DPM (Pmax)	99%	−34%
Both (Rmax)	46%	47%
Both (ave)	61%	45%
Both (Pmax)	99%	34%

Table V. HotSpot input parameters.

R convection	1.0 K/W
C convection	140.4 J/K
T ambient	45 C
Chip thickness	0.5 mm

We compare the failure rates in series and parallel topologies for the same two systems (INIT and OPT) in Figure 8. While the failure rate per core in INIT is higher, if the system is active parallel where one of the cores is redundant, the system failure rate becomes much lower. This can be observed in the trace "INIT_parallel" in the figure. OPT has two cores with non-overlapping active times. "OPT_series" demonstrates such a system of two cores in a producer–consumer pipeline, where both cores are essential for system execution. Even though the individual core failure rates are higher for INIT, it ends up with better system reliability due to its built-in redundancy. The best reliability among these systems and topologies can be reached with having the workload scheduling of OPT with built-in redundancy in the system instead of a producer–consumer pipeline, which is given in "OPT_parallel." This comparison of topologies emphasizes the positive effect of redundancy on reliability.

Another interesting problem is determining the most reliable system configuration given the following scenarios with redundancy:

(1) Having one active and one sleeping component (standby-parallel);

(2) Having the both cores processing (active-parallel) at a lower core utilization rate than the single active core.

Figure 9 compares two systems (STBY and ACTV) to highlight this discussion. The failure rate change of the standby-parallel system, STBY, is calculated with respect to having the both components active with identical

Table VI. Performance penalty.

Trace	DPM penalty	DVS penalty
Trace 1	0.2%	0.012%
Trace 2	0.67%	0.047%

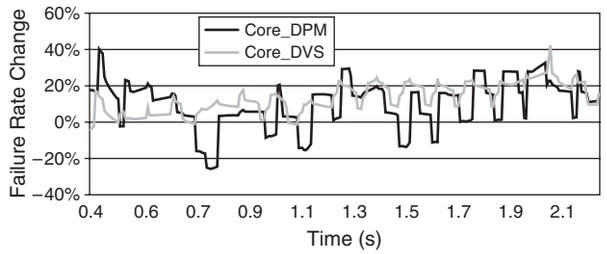

Fig. 7. 2-core system, Core0 with DPM, Core1 with DVS; both running a higher utilization trace.

workload traces; which means it does half of the work of the baseline system. ACTV has two cores with a non-overlapping workload distribution scheme as in Figure 6. While STBY has one power-managed active core and the other core in sleep state, ACTV contains two power-managed cores each running at lower utilization ratios than the single active core. Even though standby system keeps the sleeping component with very low failure rate, the active core is prone to more peaks in failure rate, which can cause a system failure in the long run.

The cycle-accurate simulations emphasize the balance that needs to be maintained between power management and temperature aware scheduling. Increasing the SoC reliability requires avoidance of fast thermal cycles through correct selection of policies while considering design choices such as the system topology and floorplan as well.

6.2. Statistical Model Results

We illustrate the tradeoff between DVS and DPM on an example of Intel's XScale processor PXA270.[7] Its power state characteristics are given in Table III. The failure rates are calculated using equations given in Section III and based on values measured for 95 nm technology from our industry partner (not Intel). We use one sleep state and four frequency settings for active and idle states, for a total of eight additional states. The workload is obtained by collecting a data trace during one day (12 hrs) of typical usage. The trace consists of standard applications—MPEG4 video, MP3 audio, WWW, email, telnet. In the simulator we implemented the "ideal" DVS policy—the policy sets the best voltage/frequency setting for each application as determined by prior analysis. In this way we compare as the best possible scenario with DVS relative to a more realistic situation with DPM policies. DPM policy is obtained by minimizing system energy consumption under the performance constraint (we solve the LP shown in Eq. (10) with no reliability constraint).

Table IV shows the percent of power savings and the percent increase in mean time to failure (MTTF) for four categories cases: no power management, only DVS, only DPM, and both. We also show results for DPM when reliability improvement is maximized (Rmax), power savings are maximized (Pmax) and the average case (Ave). DVS gives reasonable power savings—35%, with 42% improvement in MTTF. DPM has much larger power savings—up to 99%, but also causes an average 12% decline in MTTF due to on chip thermal cycles resulting from DPM. The best power savings and improvement in reliability are when DVS and DPM are combined (DVS, PM). Interestingly, when no DVS is used the difference in improvement in terms of MTTF is almost negligible between the two cases of DPM—the one where active state frequency of operation is set to maximum ($P = \max$) and the one where it is set to average ($P = \text{ave}$). Figure 11 shows that there is a clear optimal point in terms of MTTF as a function of DPM. Thus there is a need to optimize SoC reliability along with power consumption and performance. We next examine the results of optimization for a single core, followed by optimization of an SoC.

6.3. Optimization

We use a multi-processor SoC shown in Figure 10 for most of our experiments in this section. Each core's power and performance characteristics come from the datasheets[18–22] and are summarized in Table II. The cores support multiple power modes (active, idle, sleep, and off). Transition times between active and sleep state are defined by t_{ts} and t_{ta}. Reliability rates for each failure mechanism (EM, TDDB, TC) are based on measurements obtained for 95 nm technology. The failure rates are scaled depending on the current temperature of the core, which is directly affected by the core's power state and the workload. Details of failure

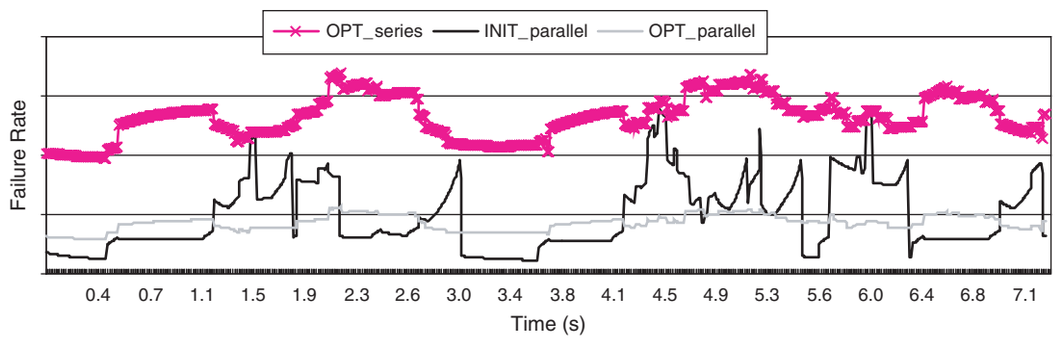

Fig. 8. System failure rates of two systems, INIT and OPT, assuming various system topologies.

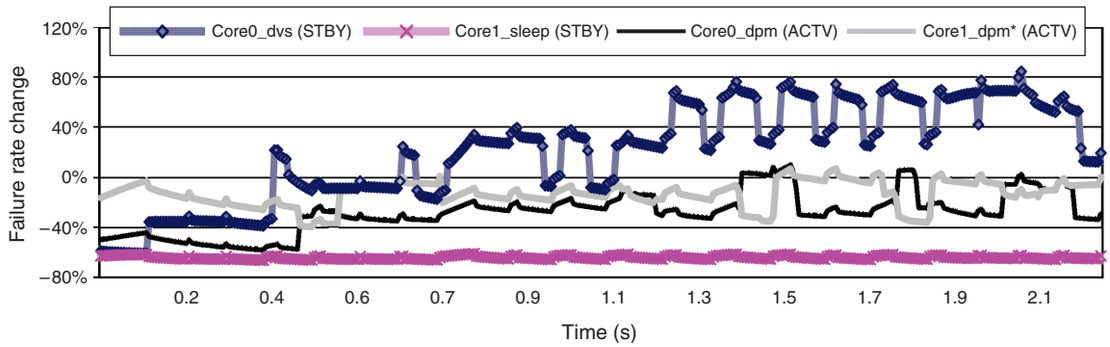

Fig. 9. Comparison of standby-parallel (STBY) and active parallel (ACTV).

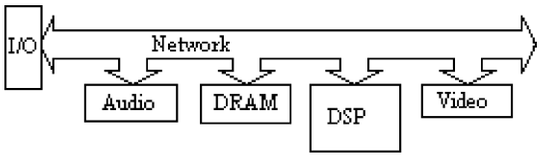

Fig. 10. System on a chip.

rate calculations have been discussed in Section 3. Each of the cores in the system is designed to meet MTTF of 10 years. Core's workload and data consumption rates ($\varphi_{workload}$) and (φ_{core_fi}) are extracted from a data trace collected during one day (12 hrs) of typical usage. Power and performance characteristics of cores are shown in Table II.

6.3.1. Single Core Optimization

We optimize the power consumption of each core listed in Table II while keeping the minimum lifetime requirement at 10 years. The objective is to observe how cores built using the same 95 nm technology and with comparable area but different power consumption respond to DPM. Optimization is performed at two internal chip temperatures (50 and 90 °C) in order to set the die operating points close to those defined in datasheets.[18–22] The optimization results for maximum power savings achievable at a specified temperature given MTTF constraint of 10 years are shown in Figure 12. At 50 °C most of the cores react positively to DPM and allow the maximum power savings

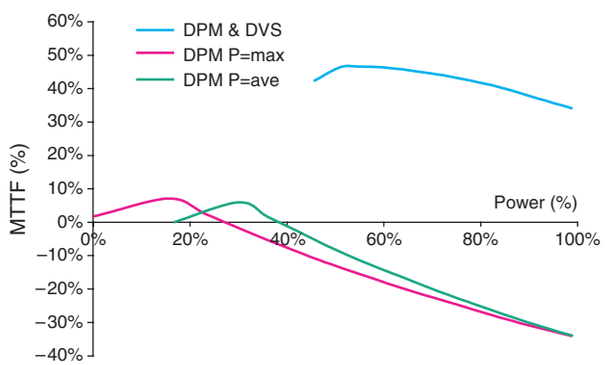

Fig. 11. Power and MTTF improvement due to DVS and DPM on XScale.

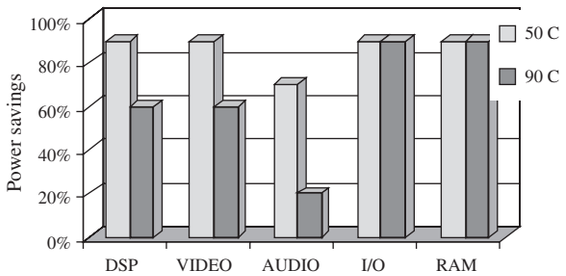

Fig. 12. Optimization of single cores.

to be achieved. When active core temperature increases to 90 °C, Figure 12 shows that maximum power savings achievable under MTTF constraint decrease due to thermal cycles for DSP, Video, and Audio cores. One way to address the problem of reduced MTTF is by either using spare cores at a large cost in area (HW spares), or migrating computation of a failed core onto one of the other available cores with significant performance degradation (SW spares).

6.3.2. SoC Optimization

Now we examine the influence of redundant components to the overall system reliability. We use the SoC shown in Figure 10 with the core parameters given in Table II. Since all cores are essential to the correct SoC operation, the initial reliability network is their series combination. Unfortunately, although each core meets MTTF requirement of 10 years, the overall system does not. To mitigate this problem we use two types of redundancy: standby sleep configuration, where currently unused cores are in a sleep state until needed, and standby off, with unused cores turned off. Since typical embedded systems do not use all of the computational resources available in SoC at all times, it is likely that some resources are at least part of the time in a low power state, and thus might be available when a failure occurs. Figure 13 and Figure 14 show the maximum power savings achievable per each core assuming system MTTF of 10 years. Clearly the best power savings are with standby off model. However, this model also has the largest wakeup delay for unused

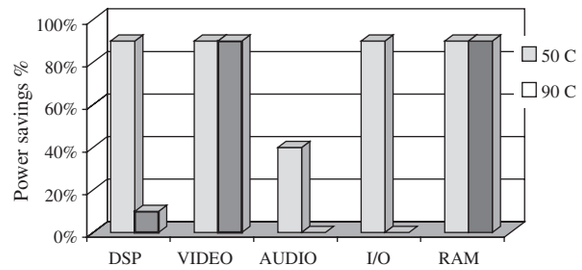

Fig. 13. Standby off redundancy model.

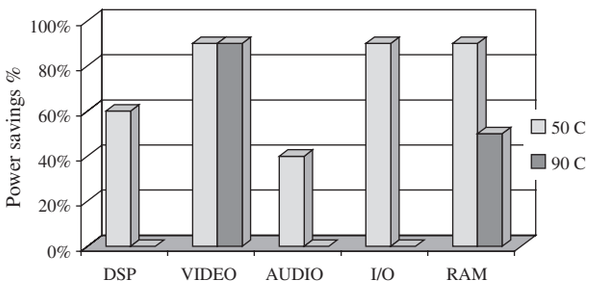

Fig. 14. Standby sleep redundancy model.

components. The standby sleep model shown in Figure 14 gives more moderate power savings but has faster activation time. Results for both models show that not all cores can operate reliably at the highest temperature (e.g., no power savings for AUDIO core at 90 °C show that the system reliability constraint of 10 years is not met). When we allow additional spares for DSP, AUDIO, and I/O in standby off mode, then the overall system meets MTTF of 10 years while getting power savings of 40%.

7. CONCLUSION

In this work, we describe a comprehensive methodology for analysis of multi-processor SoC reliability. We show that aggressive power management can cause significant impact on reliability, and we introduce two simulation/optimization methodologies to analyze and overcome this problem. The cycle-accurate model observes the changes in failure rates due to short-time horizon workload and policy changes. This way, adaptive strategies can be implemented for controlling the adverse effects of power management on reliability. As an example, we show that reliability aware workload scheduling can maintain a low and stable temperature and failure rate.

We introduce statistical simulation that explores the power reliability trade-offs over the lifetime of the device. In addition, we present a first optimization methodology for determining the best power management policy that can meet both system reliability (MTTF) and performance constraints. We show that with our optimization the system can meet its performance and reliability constraints while saving more than 40% in power consumption.

References

1. S. Gunther, F. Binns, D. M. Carmean, and J. C. Hall, Managing the impact of increasing microprocessor power consumption. *Intel Technology Journal* **(2001)**.
2. K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, Temperature-aware microarchitecture. *Proceedings of International Symposium on Computer Architecture*, **(2003)**, pp. 1–12.
3. L. Shang, L. Peh, A. Kumar, and N. Jha, Thermal Modeling, characterization and management of on-chip networks. *Proceedings of the 37th Annual IEEE/ACM International Symposium on Microarchitecture* **(2004)**, pp. 67–78.
4. J. Srinivasan and S. Adve, Predictive dynamic thermal management for multimedia applications. *Proceedings of the 17th Annual International Conference on Supercomputing* **(2003)**, pp. 109–120.
5. J. Srinivasan, S. Adve, P. Bose, and J. Rivers, Exploiting structural duplication for lifetime reliability enhancement. *Proceedings of the 32nd Annual International Symposium on Computer Architecture*, **(2005)**, pp. 520–531.
6. T. Simunic, K. Mihic, and G. De Micheli, Reliability and power management of integrated systems. *Proceedings of the Digital System Design* **(2004)**, pp. 5–11.
7. Intel PXA270 Datasheets, www.intel.com.
8. P. Shivakumar, S. W. Keckler, C. R. Moore, and Doug Burger, Exploiting microarchitectural redundancy for defect tolerance. *Proceedings of the 21st International Conference on Computer Design* **(2003)**, p. 481.
9. T. Simunic and S. Boyd, Managing power consumption in networks on chips. *Proceedings of the Design, Automation, and Test in Europe* **(2002)**, pp. 110–116.
10. T. Simunic, L. Benini, P. Glynn, and G. De Micheli, Event-driven power management. *IEEE Transactions on CAD* **(2001)**, pp. 840–857.
11. J. Srinivasan, S. V. Adve, P. Bose, J. Rivers, and C. K. Hu, RAMP: A Model for reliability aware microprocessor design. *IBM Research Report*, RC23048 **(2003)**.
12. K. Skadron, T. Abdelzaher, and M. R. Stan, Control-theoretic techniques and thermal-RC modeling for accurate localized dynamic thermal management. *Proceedings of the 8th International Symposium on High-Performance Computer Architecture* **(2002)**, p. 17.
13. Semiconductor device reliability failure models. *International Sematech Technology Transfer Document* 00053955A-XFR **(2000)**.
14. H. V. Nguyen, Multilevel Interconnect Reliability on the Effects of Electro-Thermomechanical Stresses Ph.D. dissertation, University of Twente, Netherland **(2004)**.
15. M. Huang and Z. Suo, Thin film cracking and ratcheting caused by temperature cycling. *J. Mater. Res.* **(2000)**, Vol. 15, p. 1239.
16. R. Degraeve, J. L. Ogier, R. Bellens, P. J. Roussel, G. Groeseneken, and H. E. Maes, A New model for the field dependence of intrisic and extrinsic time-dependent dielectric breakdown. *IEEE Transactions on Electron Devices* **(1998)**, Vol. 45, pp. 472–481.
17. E. E. Lewis, Introduction to Reliability Engineering, Wiley **(1996)**.
18. TMS320C6211, TMS320C6211B, Fixed-point digital signal processors. Texas Instruments **(2002)**.
19. SAF7113H datasheet. *Philips Semiconductors* **(2004)**.
20. SST-Melody-DAP audio processor. *Analog Devices* **(2002)**.
21. MSP430×11×2, MSP430×12×2 mixed signal microcontroller. Texas Instruments **(2004)**.
22. RDRAM 512Mb. *Rambus* **(2003)**.
23. A. Maheshwari, W. Burleson, and R. Tessier, Trading off reliability and power-consumption in ultra-low power systems. *Proceedings of the International Symposium on Quality Electronic Design* **(2002)**, pp. 361–366.
24. P. Stanley-Marbell and D. Marculescu, Dynamic fault-tolerance and metrics for battery powered, failure-prone systems. *Proceedings of*

the International Conference on Computer-Aided Design **(2003)**, p. 633.

25. P. Pillai and K. G. Shin, Real-time dynamic voltage scaling for low-power embedded operating systems. *Proceedings of the 18th ACM Symposium on Operating Systems Principles* **(2001)**, pp. 89–102.

26. L. Benini, A. Bogliolo, and G. De Micheli, A survey of design techniques for system-level dynamic power management. *IEEE Transactions on VLSI* **(2000)**, Vol. 8, pp. 299–316.

27. E.-Y. Chung, L. Benini, A. Bogliolo, Y.-H. Lu, and G. De Micheli, Dynamic power management for nonstationary service requests. *IEEE Transactions on Computers* **(2002)**, Vol. 51, pp. 1345–1361.

28. K. Choi, R. Soma, and M. Pedram, Dynamic voltage and frequency scaling based on workload decomposition. *Proceedings of the International Symposium on Low Power Electronics and Design* **(2004)**, pp. 174–179.

29. Todd M. Austin, DIVA: A reliable substrate for deep submicron microarchitecture design. *Proceedings of the 32nd Annual ACM/IEEE International Symposium on Microarchitecture* **(1999)**, pp. 196–207.

30. T. Simunic, K. Mihic, and G. De Micheli, Optimization of reliability and power consumption in systems on a chip. *Proceedings of the International Workshop on Power and Timing Modeling, Optimization, and Simulation* **(2005)**, pp. 237–246.

31. A. Maxiaguine, Y. Liu, S. Chakraborty, and W. T. Ooi, Identifying "representative" workloads in designing MpSoC platforms for media processing. *Proceedings of the Workshop on Embedded Systems for Real-Time Multimedia* **(2004)**.

32. L. Benini, D. Bertozzi, A. Bogliolo, F. Menichelli, and M. Olivieri, MPARM: Exploring the multi-processor SoC design space with system C. *Springer J. of VLSI Signal Processing* **(2005)**, Vol. 41, pp. 169–182.

Ayse Kivilcim Coskun

Ayse Kivilcim Coskun *is currently a Ph.D. student in Computer Science and Engineering Department at UC San Diego. She received her B.S. degree in Microelectronics Engineering from Sabanci University, Turkey in 2003, together with a minor degree in Physics. Her research interests are reliability and power management modeling and optimization in multi-core SoCs, and fault tolerant computer architectures. Her advisor in UCSD is Tajana Simunic-Rosing. She worked as a summer intern at Integrated Systems and Microelectronic Systems Laboratories in summer 2005, working with Dr. Giovanni De Micheli and Dr. Yusuf Leblebici.*

Tajana Šimunić Rosing

Tajana Šimunić Rosing *is currently an Assistant Professor in Computer Science Department at UCSD. Her research interests are low-power system design, embedded systems and wireless system design. Prior to this she was a full time researcher at HP Labs while working part-time at Stanford University. At Stanford she has been involved with leading research of a number of graduate students and has taught graduate level classes. She finished her Ph.D. in 2001 at Stanford University, concurrently with finishing her Masters in Engineering Management. Her Ph.D. topic was Dynamic Management of Power Consumption. Prior to pursuing the Ph.D., she worked as a Senior Design Engineer at Altera Corporation. She obtained the M.S. in EE from University of Arizona. Her M.S. thesis topic was high-speed interconnect and driver-receiver circuit design. She has served at a number of Technical Paper Committees, and is currently an Associate Editor of IEEE Transactions on Circuits and Systems.*

Kresimir Mihic

Kresimir Mihic *is currently a product engineer in Magma Design Automation in Santa Clara, CA. He started his Ph.D. in Stanford University in 2003. His research interest's are SoC reliability and dynamic power management, and distributed signal processing. Prior to his current position in Magma, he worked in Cypress Semiconductors as a process engineer between 2000–2005.*

Giovanni De Micheli

Giovanni De Micheli *is Professor and Director of the Integrated Systems Center at EPF Lausanne, Switzerland, and President of the Scientific Committee of CSEM, Neuchatel, Switzerland. Previously, he was Professor of Electrical Engineering at Stanford University. He held positions at the IBM T. J. Watson Research Center, Yorktown Heights, New York, at the Department of Electronics of the Politecnico di Milano, Italy and at Harris Semiconductor, Melbourne, Florida. He holds a Nuclear Engineer degree (Politecnico di Milano, 1979), a M.S. and a Ph.D. degree in Electrical Engineering and Computer Science (University of California at Berkeley, 1980 and 1983). His research interests include several aspects of integrated circuits and systems design, with particular emphasis on synthesis, system-level design, hardware/software co-design, and low-power design. He is author of "Synthesis and Optimization of Digital Circuits," McGraw-Hill, 1994, co-author and/or co-editor of five other books and of over 300 technical articles. He has been member of the technical advisory board of several companies, including Magma Design Automation, Coware, Aplus Design Technologies, Ambit Design Systems, and ST Microelectronics. Dr. De Micheli is the recipient of the 2003 IEEE Emanuel Piore Award for contributions to computer-aided synthesis of digital systems. He is a Fellow of ACM and IEEE. He received the Golden Jubilee Medal for outstanding contributions to the IEEE CAS Society in 2000. He received the 1987 D. Pederson Award for the best paper on the IEEE Transactions on CAD/ICAS, two Best Paper Awards at the Design Automation Conference, in 1983 and in 1993, and a Best Paper Award at the DATE Conference in 2005. He was President of the IEEE CAS Society in 2003, and he is currently President Elect of the IEEE Council on EDA. He was Editor in Chief of the IEEE Transactions on CAD/ICAS in 1987–2001. Dr. De Micheli was the Program Chair and General Chair of the Design Automation Conference (DAC) in 1996–1997 and 2000 respectively. He was the Program and General Chair of the International Conference on Computer Design (ICCD) in 1988 and 1989 respectively. He was also co-director of the NATO Advanced Study Institutes on Hardware/Software Co-design, held in Tremezzo, Italy, 1995 and on Logic Synthesis and Silicon Compilation, held in L'Aquila, Italy, 1986. He is a founding member of the ALaRI institute at Universita' della Svizzera Italiana (USI), in Lugano, Switzerland.*

Yusuf Leblebici

Yusuf Leblebici *received the B.S. and M.S. degrees in electrical engineering from Istanbul Technical University in 1984 and 1986, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC) in 1990. Between 1991 and 2001, he worked as a faculty member at UIUC, at Istanbul Technical University, and at Worcester Polytechnic Institute (WPI), where he established and directed the VLSI Design Laboratory, and also served as a project director at the New England Center for Analog and Mixed-Signal IC Design. He also worked as the Microelectronics Program Coordinator at Sabanci University. Since January 2002, Dr. Leblebici is a full professor at the Swiss Federal Institute of Technology in Lausanne (EPFL), and director of the Microelectronic Systems Laboratory. His research interests include design of high-speed CMOS digital and mixed-signal integrated circuits, computer-aided design of VLSI systems, modeling, and simulation of nano-electronic circuits, intelligent sensor interfaces, and VLSI reliability analysis. Dr. Leblebici is the coauthor of two textbooks, "Hot-Carrier Reliability of MOS VLSI Circuits" (Kluwer Academic, 1993) and "CMOS Digital Integrated Circuits: Analysis and Design" (McGraw Hill, 1996, 1999, and 2003), as well as numerous scientific articles published in international journals and conferences. He has served on the organizing and steering committees of the 1995 European Conference on Circuit Theory and Design (ECCTD 1995) and the 2004 European Workshop on Microelectronics Education (EWME 2004). He is the Co-Chairman of the Joint 2006 European Solid-State Circuits/Device Research Conference (ESSCIRC/ESSDERC) to be held in Montreux, Switzerland. Dr. Leblebici served as an Associate Editor of IEEE Transactions on Circuits and Systems II between 1998 and 2000, and as an Associate Editor of IEEE Transactions on VLSI between 2001 and 2003. He received the NATO Science Fellowship award in 1986, he has been an Honors Scholar of the Turkish Scientific and Technological Research Council in 1987–1990, and he received the Young Scientist Award of the same Council in 1995. In 1999, Dr. Leblebici was the recipient of the Joseph Samuel Satin Distinguished Fellow Award of the Worcester Polytechnic Institute.*