Characterizing the Lifetime Reliability of Manycore Processors with Core-Level Redundancy

Lin Huang and Qiang Xu
CUhk REliable Computing Laboratory (CURE)
Department of Computer Science & Engineering
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong
Email: {lhuang,qxu}@cse.cuhk.edu.hk

ABSTRACT

With aggressive technology scaling, integrated circuits suffer from ever-increasing wearout effects and their lifetime reliability has become a serious concern for the industry. For manycore processors that integrate a large number of processor cores on a single silicon die, introducing core-level redundancy is an effective way to alleviate this problem. There are, however, many strategies to make use of the redundant cores, which have different implications on the aging effects of embedded processors. How to characterize the lifetime reliability of manycore processors with different usages is therefore an important and relevant problem. In this paper, we propose a novel analytical method to tackle the above problem, which captures the impact of workloads and the associated temperature variations. We then use the proposed model to analyze the lifetime reliability for manycore processors with various redundancy configurations. Finally, the effectiveness of the proposed method is demonstrated with extensive experiments.

1. INTRODUCTION

As technology advances, industry has started to employ multiple processor cores on a single silicon die to improve performance through parallel execution. Such chip multiprocessors, also known as multicore or manycore processors (depending on the number of cores on the die), being much more power-efficient than unicore processors with extremely high frequency, have become increasingly popular [4, 6].

Due to device wearout, integrated circuits (ICs) suffer from various types of intrinsic failures, which manifest themselves after some time of operation and determine the circuits' service life. With the relentless technology scaling, the lifetime reliability of high-performance ICs has become a serious concern for the industry [10, 16, 18, 13]. Major intrinsic failures include TDDB in the gate oxides, EM in the interconnects, NBTI stresses that shift PMOS transistor threshold voltages, and thermal cycling. Many widely accepted reliability models for the above failure mechanisms have been proposed and empirically validated by academia and industry [3, 1, 2, 22]. These models, however, are not readily applicable in characterizing the lifetime reliability of manycore processors, because they assume constant temperature and voltage while these values vary significantly at runtime.

One method to obtain defect-tolerance capabilities is to incorporate redundant circuits in a system and use them as replacements when some units are faulty. This strategy can be also used to extend the service life of IC products. In particular, for manycore processors, employing core-level redundancy is a more attractive solution than introducing complex microarchitecture-level redundancy and has been practiced in the industry. There are, however, many ways to make use of the redundant cores. We can configure some cores as standbys and use them only when some of the active cores fail. We can also activate all cores from the beginning and remove the faulty cores during the systemąŕs lifetime. Moreover, we have the freedom to dynamically configure which cores to serve as active cores and which cores to serve as spares at a specific time. As ICs' wearout-related failure rates are significantly related to operational conditions such as temperature and/or voltage, these strategies result in different aging stress on processors. How to characterize the lifetime reliability of manycore processors with different usages is therefore an important and relevant problem.

To address the above problem, in this paper, we explicitly consider the temperature variations caused by workloads in our analytical model to estimate the lifetime reliability of manycore processors with various redundancy schemes. To be specific, we introduce a parameter namely *wearout rate* to reflect a core's aging effect in its different operational state, which is computed with the temperature distribution of the core. We then model the lifetime reliability of manycore processors using wearout rates. Finally, extensive experiments are conducted to compare manycore processors in terms of both lifetime reliability and performance, under various workloads, service time distributions, and redundancy configurations.

The remainder of this paper is organized as follows. Section 2 reviews related prior work and motivates this paper. Our proposed analytical model for the lifetime reliability of a processor core is then detailed in Section 3 and we use this model to investigate the impact of various redundancy schemes on the service life of manycore processors in Section 4. Next, Section 5 and Section 6 present our experimental methodology and experimental results, respectively. Finally, Section 7 concludes this work.

2. RELATED WORK AND MOTIVATION

Processor lifetime reliability is significantly affected by its operating conditions, which vary with different applications running on the processor. In [16, 17], Srinivasan *et al.* proposed an application-aware architecture-level model, namely *RAMP* model, which is able to dynamically track lifetime reliability of a processor according to changes in application behavior. Later, Shin *et al.* [14] defined reference circuits and introduced a structure-aware model that takes the vulnerability of basic structures of the microarchitecture (e.g., register files, latches and logic) to different types of failure mechanisms into account. Both Srinivasan's and Shin's models target unicore architecture. Coskun *et al.* [5] introduced two analytical frameworks for the lifetime reliability of multicore systems: a cycle-accurate simulation methodology and a statistical one, assuming uniform device density.

Some of the above models (e.g.,[16, 5]) assumed exponential failure distributions (i.e., constant failure rate) and thus cannot capture the processors' accumulated aging effects. In practice, we expect increasing failure rates as systems grow older and it is suggested to use non-exponential distributions, such as Weibull distribution and/or lognormal distribution, to describe the influence of hard errors [17, 1, 20]. Consider NBTI as an example, the increase of threshold voltage ΔV_{th} at time t highly depends on the usage history of the transistors. In [17], the authors modeled processors with microarchitecture-level redundancy as series-parallel failure systems with lognormal failure distribution and used a simple MIN-MAX analysis to determine the system lifetime. This model, however, is not applicable for analyzing the lifetime reliability of manycore processors with core-level redundancy. It cannot reflect the load-sharing feature of manycore processors. More importantly, series-parallel model is not applicable for many often-used configurations, such as standby redundant system, wherein some cores start their service life only when permanent core failures occur in the system.

Recently, Huang and Xu [9, 11] developed a high-level analytical model for the lifetime reliability of manycore processors, which takes arbitrary failure distribution and load-sharing feature into account. In this work, a processor core is assumed to be in three possible states: *processing state*, *wait state*, and *spare state*, each corresponding to a unique failure distribution. The above assumption, however, oversimplifies this problem because the lifetime reliability of a processor core highly depends on its operational temperature, which varies with different applications running on it. That is, even if two cores are in the same states and have the same usage history, they do not necessarily have the same failure rates. From this aspect, we need to extend the discrete states into a series of continuous states for more accurate estimation of the system's lifetime reliability, by taking the temperature and structural information that affect the system's lifetime reliability into account.

The above observations have motivated the work studied in this paper.

3. PROPOSED ANALYTICAL MODEL FOR THE LIFETIME RELIABILITY OF PROCESSOR CORES

As discussed above, the circuit wearout effects are related to its operational status such as temperature, voltage and frequency, which are not explicitly considered in the analytical model in [9, 11]. In this section, we first examine the impact of these factors on processor cores' lifetime reliability. Next, we consider the impact of workloads by mapping them into the different temperature distributions of processor cores.

3.1 Impact of Temperature, Voltage, and Frequency

To examine the impact of temperature, supply voltage, and clock frequency on the wearout effect of a single processor core, we start with the simplest case: no failures occur in the system up to time t. Under such circumstances, the task interarrival time distribution of a core is fixed up to time t.

We use the notation $\mathcal{R}(t,\theta)$ to denote a general reliability function, where θ represents the general scale parameter by which time t is divided and depends on temperature and processor's execution mode. For example, the commonly-used Weibull failure distribution has the form $\mathcal{R}_{Weibull}(t,\theta)=e^{-(\frac{t}{\theta})^{\beta}}$. Without ambiguity, we hereafter drop the notation θ because of its generality and refer to the general reliability function as $\mathcal{R}(t)$. Note that, $\mathcal{R}(t)$ does not necessarily to be an exponential distribution.

Without loss of generality, we consider a core can be in any state s of set $\mathcal{S}$. An example of set $\mathcal{S}$ is defined in [9], namely, $\{processing, wait, spare\}$. Depending on a core's state, temperature, voltage, and frequency, let a subdivision of $[0,t]$ be a finite sequence $0=\tilde{t}_0<\tilde{t}_1<\cdots<\tilde{t}_d=t$, which partitions the interval $[0,t]$ into d sub-intervals. In each time sub-interval $[\tilde{t}_j,\tilde{t}_{j+1}]$ (referred as sub-interval j hereafter), the core remains in the same state, and its voltage and frequency keep unchanged. We denote the state, voltage, and frequency in sub-interval j as s_j, V_j and f_j, respectively. In addition, the temperature variation in every sub-interval is very small. Formally, since temperature is function of time, we use $T(t)$ represent a core's temperature at time t. For all $\varepsilon>0$ there exists $\delta>0$ such that, if the largest partition $\max_j(\tilde{t}_{j+1}-\tilde{t}_j)<\delta$ then for all j the difference between $\max_{\tilde{t}_j<t<\tilde{t}_{j+1}}T(t)$ and $\min_{\tilde{t}_j<t<\tilde{t}_{j+1}}T(t)$ is less than ε. Denote by $\Delta_j\tilde{t}$ the difference $\tilde{t}_{j+1}-\tilde{t}_j$ and by T_j^* any value of T such that $\min_{\tilde{t}_j<t<\tilde{t}_{j+1}}T(t)\le T_j^*\le\max_{\tilde{t}_j<t<\tilde{t}_{j+1}}T(t)$. The corresponding scale parameter is expressed as $\theta_{s_j}(T_j^*,V_j,f_j)$.

By the definition of reliability function, we express $R(t)$ in $[\tilde{t}_0,\tilde{t}_1]$ as

$$R(\tau)=\mathcal{R}\big(\frac{\theta}{\theta_{s_0}(T_0^*,V_0,f_0)}\cdot(\tau-\tilde{t}_0)\big),\qquad \tilde{t}_0\le\tau\le\tilde{t}_1 \tag{1}$$

Next, substituting $\tau=\tilde{t}_1$ into Eq. (1) yields the core's reliability at the end of the first sub-interval. Because of the continuity of reliability function, the reliability at the beginning of the second sub-interval has the same value as Eq. (2). With this condition, we express the reliability in the second sub-interval $[\tilde{t}_1,\tilde{t}_2]$ as

$$R(\tau)\;=\;\mathcal{R}\big(\frac{\theta}{\theta_{s_1}(T_1^*,V_1,f_1)}\cdot(\tau-\tilde{t}_1)+\frac{\theta}{\theta_{s_0}(T_0^*,V_0,f_0)}\cdot(\tilde{t}_1-\tilde{t}_0)\big),\qquad \tilde{t}_1\le\tau\le\tilde{t}_2 \tag{2}$$

By the same argument and the limiting process, at time t we have

$$R(t)=\mathcal{R}\big(\theta\cdot\lim_{\substack{d\to\infty\\ \max\Delta_j t\to 0}}\sum_{j=0}^{d-1}\frac{1}{\theta_{s_j}(T_j^*,V_j,f_j)}\cdot\Delta_j\tilde{t}\big) \tag{3}$$

In this expression, the state parameter s_j is in the set $\mathcal{S}$ for any j. To simplify Eq. (3), we introduce a filter function over time horizon ${}_s(\tilde{t},V,f)$ such that it equals 1 if the core is in state s with voltage V and frequency f at time $\tilde{t}$ while 0 otherwise. With this notation, Eq. (3) comes down to

$$R(t)=\mathcal{R}\big(\theta\cdot\sum_{s\in\mathcal{S}}\sum_V\sum_f\int_0^t\frac{{}_s(\tilde{t},V,f)}{\theta_s(T,V,f)}\mathrm{d}\tilde{t}\big) \tag{4}$$

In this equation, we integrate $\frac{1}{\theta_s(T,V,f)}$ over $\tilde{t}$. To integrate over $\mathrm{d}T$ (T is a function of $\tilde{t}$), we denote by $\psi_s(T,V,f)\mathrm{d}T$ the accumulated time in state s with voltage V and frequency f in an infinitesimal temperature interval $\mathrm{d}T$ around T. We use it to substitute $\mathrm{d}\tilde{t}$ and change lower and upper limits of integration accordingly, yielding

$$R(t)=\mathcal{R}\big(\theta\cdot\sum_{s\in\mathcal{S}}\sum_V\sum_f\int_0^\infty\frac{1}{\theta_s(T,V,f)}\cdot\psi_s(T,V,f)\cdot\mathrm{d}T\big) \tag{5}$$

Further, we use $\nu_s(T,V,f)$ to represent the probability density function (p.d.f.) of a core with temperature T, given the core is in state s. Also, π_s is defined as the probability a core being in state s. Thus, the fraction of accumulated time within which the core falls in a infinitesimal interval $\mathrm{d}T$ at T and is in state s can be approximated by $\pi_s\cdot\nu_s(T,V,f)\cdot\mathrm{d}T$. Hence, Eq. (5) can be rewritten as

$$R(t)=\mathcal{R}\big(\theta\cdot\sum_{s\in\mathcal{S}}\sum_V\sum_f\int_0^\infty\frac{1}{\theta_s(T,V,f)}\cdot\pi_s\cdot\nu_s(T,V,f)\cdot t\cdot\mathrm{d}T\big) \tag{6}$$

Because π_s and t are independent of V, f and T, they can be moved outside of the corresponding integral and summation signs to obtain

$$R(t)=\mathcal{R}\Big(\theta\cdot\sum_{s\in\mathcal{S}}\pi_s\cdot\big(\sum_V\sum_f\int_0^\infty\frac{1}{\theta_s(T,V,f)}\cdot\nu_s(T,V,f)\mathrm{d}T\big)\cdot t\Big) \tag{7}$$

Now, we are ready to introduce the formal definition of *wearout rate* in state s, a quantity that describes the rate of core suffering from wearout effects, namely,

$$\Omega_s\equiv\sum_V\sum_f\int_0^\infty\frac{\nu_s(T,V,f)}{\theta_s(T,V,f)}\mathrm{d}T \tag{8}$$

Using Ω_s, Eq. (7) can be rewritten as

$$R(t)=\mathcal{R}\big(\theta\cdot\sum_{s\in\mathcal{S}}\pi_s\cdot\Omega_s\cdot t\big) \tag{9}$$

Clearly, ν_s follows a constraint that

$$\sum_V\sum_f\int_0^\infty\nu_s(T,V,f)\mathrm{d}T=1 \tag{10}$$

Here, $\nu_s(T,V,f)$ is the conditional p.d.f. of temperature T, voltage V, and frequency f with given state s. According to the theorem of total probability, it is possible for us to drop s from notation Ω_s and express wearout rate in a concise form. As both θ and t are independent of wearout rate, from Eq. (9) we define

$$\Omega=\sum_{s\in\mathcal{S}}\pi_s\cdot\Omega_s=\int_0^\infty\sum_{s\in\mathcal{S}}\sum_V\sum_f\frac{\pi_s\cdot\nu_s(T,V,f)}{\theta_s(T,V,f)}\mathrm{d}T \tag{11}$$

Thus, Eq. (9) can be written as

$$R(t)=\mathcal{R}\big(\theta\cdot\Omega\cdot t\big) \tag{12}$$

In particular, if the core has the same frequency and voltage in various states other than in the cold standby mode, we can redefine scale parameter $\theta(T)$ according to these parameters. Since a core in cold standby state is switched off, its lifetime is close to infinity, i.e., $\theta_{spare}(T)\to\infty$. In other words, the wearout rate contributed by this state is approximated to zero. Therefore, we are only interested in the temperature distribution given the core is not in cold standby, denoted as $\nu(t)$. For a core which is not set into cold standby state within a time interval, from Eq. (11), we have

$$\Omega=\int_0^\infty\frac{\nu(T)}{\theta(T)}\mathrm{d}T \tag{13}$$

where, temperature distribution $\nu(T)$ follows

$$\int_0^\infty\nu(T)\mathrm{d}T=1 \tag{14}$$

3.2 Impact of Workloads

In many systems, the workload distribution of a core is not fixed. For instance, in a gracefully degrading manycore processor with redundant cores, all cores share the workload initially. We therefore examine how workloads affect the wearout rate in this section.

Remind the mathematical derivation of unified reliability function is independent of workload distribution. Suppose a set $\mathcal{M}=\{1,2,\cdots,m\}$ of cores equally share the workload, the probability for core i ($1\le i\le m$) to process any task is $\frac{1}{m}$. Thus, given the workload distribution of the entire system, it is easy to know every core's workload. In our model, it is reflected in temperature distributions $\nu_s(T,V,f)$ and hence the wearout rate Ω. Fig. 1 shows typical temperature distributions of a core under various workloads ρ^{sys} (the formal definition is introduced in Section 5.1). The data is collected from HotSpot [15] for an application flow composed of 10,000 tasks. Without loss of generality, we assume every state s corresponds to a single supply voltage value V and clock frequency value f in this numerical experiment. We add a subscript m to indicate the quantity of cores which process workloads in the system. For example, Ω_{36} can be used to represent the wearout rate of system that contains 36 active units.

We then present how to extend the definition of Ω, which is drawn from the expression of $\mathcal{R}(t)$ assuming the distribution is fixed from time zero up to t, to compute wearout rate Ω_m for any active core quantity m. Under the same usage strategy, for the same system the difference in wearout rate caused by different workload is reflected in temperature distributions $\nu_{s,m}(T,V,f)$ over T and the probabilities of a core being in various states $\pi_{s,m}$. Consequently, from Eq. (11) we have

$$\Omega_m = \int_0^\infty \sum_{s\in S}\sum_V\sum_f \frac{\pi_{s,m}\cdot \nu_{s,m}(T,V,f)}{\theta_s(T,V,f)}\mathrm{d}T \tag{15}$$

Even if a core has experienced other workload distribution before the current one, Eq. (15) is able to capture the aging effect in this time interval. To take an example, suppose all n cores in a system equally share workload at the beginning, then one of them fails at time t_1 resulting in a heavier load on every surviving core. In this case, we can use Ω_n and Ω_{n-1} to represent the wearout rate in two states respectively. From Eq. (12), the reliability of a surviving core at time $\hat{t}$ ($\hat{t} \le t_1$) is $R(\hat{t}) = \mathcal{R}(\theta\cdot\Omega_n\cdot\hat{t})$. Then, we analyze the second state. Since this core enters its second state at time t_1, its initial reliability of the second state is $R(t_1) = \mathcal{R}(\theta\cdot\Omega_n\cdot t_1) = \mathcal{R}(A_1)$, where $A_1 \equiv \theta\cdot\Omega_n\cdot t_1$. Sequentially, by the similar argument with that in Section 3.1, we can express the reliability of a surviving core at time t ($t > t_1$) as $R(t) = \mathcal{R}\big(\theta\cdot\Omega_{n-1}\cdot(t-t_1)+A_1\big)$.

4. LIFETIME RELIABILITY ANALYSIS FOR MANYCORE PROCESSORS WITH VARIOUS REDUNDANCY SCHEMES

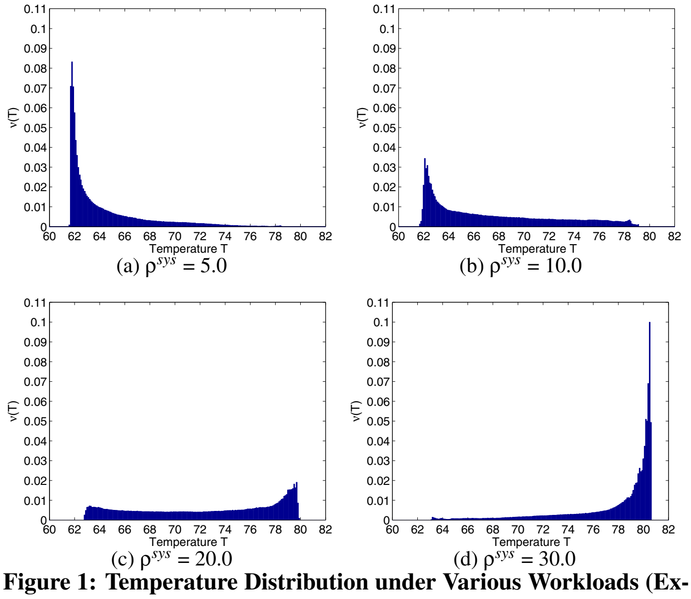
Modeling the lifetime reliability of manycore processors with redundancy is more complicated. This is because, the status, workload and the corresponding failure rate of each core in a system can be time-varying, depending on the redundancy configuration and wearout-related failure occurrences. In this section,we focus on three redundant schemes and discuss their lifetime reliability models in detail. Then, we present how to extend the proposed model for heterogeneous manycore systems.

4.1 Gracefully Degrading System (GDS)

In GDS, initially all n cores are configured as active units. When a core fails, the system will be reconfigured in a gracefully degrading manner, that is, the remaining $(n-1)$ good cores share the system workload. This process continues until there are only k good cores left. In that situation, if one more core fails, the entire system will be considered as faulty. The number of cores sharing workloads can be n, $n-1$, $\cdots$, k, and corresponding wearout rates are Ω_n, Ω_{n-1}, $\cdots$, Ω_k, respectively. By extending deduction procedure presented in Section 3, for any surviving core at time t, given that the system contains $(n-\ell)$ good cores at t and the i^{th} permanent component failure in the system occurs at time t_i ($1\le i\le \ell$), its reliability can be expressed as

$$R^{GDS}(t|\ell) = \mathcal{R}\Big(\theta\cdot\big(\sum_{i=0}^{\ell-1}\Omega_{n-i}\cdot(t_{i+1}-t_i)+\Omega_{n-\ell}\cdot(t-t_\ell)\big)\Big) \tag{16}$$

The event that all surviving components after ℓ core failures is still functioning at time t, where $t > t_\ell$, can be modeled as a series failure system. We use $\big(R^{GDS}(t|\ell)\big)^{n-\ell}$ to represent its probability. The next step is to uncondition it by being aware that it is conditioned, that the occurrence time of past failures are assumed to be given. Similar to the system lifetime analysis in [9], as the reliability of a core given past i failures is $R^{GDS}(t|i)$, the event for its $(i+1)$ failure occurring at t_{i+1} has probability $-\frac{\mathrm{d}}{\mathrm{d}t}R^{GDS}(t|i)\Big|_{t=t_{i+1}}$, referred to as $f^{GDS}(t_{i+1}|i)$ hereafter. Therefore, denoting by $R^{GDS,sys}(t,\ell)$ the probability of event that the system has experienced exactly ℓ core failures before time t and by the theorem of total probability, we have Eq. (17) (see next page), where the domain is $\mathbf{D} = \{(t_1,\cdots,t_\ell)\in\mathbb{R}^\ell : 0 < t_1 < \cdots < t_\ell < t\}$.



(a) $\rho^{sys} = 5.0$ (b) $\rho^{sys} = 10.0$ (c) $\rho^{sys} = 20.0$ (d) $\rho^{sys} = 30.0$

Figure 1: Temperature Distribution under Various Workloads (Exponential Service Time).

Then, since the event that a GDS system is functioning can be expressed as the union of a set of mutually exclusive events, the system reliability $R^{GDS,sys}(t)$ is therefore given by

$$R^{GDS,sys}(t) = \sum_{\ell=0}^{n-k} R^{GDS,sys}(t,\ell) \tag{18}$$

Consequently, the system mean time to failure is Eq. (19).

4.2 Processor Rotation System (PRS)

Processor cores can be used in a rotation manner to balance their aging effects. That is, they operate alternatively in active mode and spare mode and spend a relatively longer period in each mode when compared to the execution time of each single task in every state. Moreover, the duration is quite small when compared to the lifetime of the system. In [13], the authors showed an example for caches enabled in a round-robin manner.

For modeling lifetime reliability, we consider a more general case that in any configuration k out of n cores serve as active ones while the remaining $(n-k)$ have no power supply. The reconfiguration is conducted every time interval T_r, which is much shorter than a core's service life (typically a few years) but much longer than a task's execution time. At every reconfiguration, the $(n-k)$ oldest cores (that is, the cores with highest age) are shut down, and all spare ones convert to active mode. From a core's point of view, before the first failure in the system, its accumulated time up to time t as active core can be approximated as $\frac{k}{n}\cdot t$. Its wearout rate under this condition should be Ω_k, because there are k active cores sharing workload. On the other hand, a core does not have power supply in the remaining $(1-\frac{k}{n})\cdot t$. Recall that the wearout rate in these time intervals is approximated to zero. Therefore, its reliability before the first component failure is given by

$$R^{PRS}(t|0) = \mathcal{R}(\theta\cdot\Omega_k\cdot\frac{k}{n}\cdot t) \tag{20}$$

Then, we generalize this expression to the case that the number of failure cores in the system can be 0, 1, $\cdots$, $(n-k)$. From t_i to t_{i+1}, the system composed of $(n-i)$ good components within which k are active at any time. Since a surviving core's accumulated time in this time interval depends on the quantity of both active cores and redundant ones, it can be approximated as $\frac{k}{n-i}\cdot(t_{i+1}-t_i)$. Its wearout rate, on the other hand, remains Ω_k. We therefore compute its reliability by

$$R^{PRS}(t|\ell) = \mathcal{R}\Big(\theta\cdot\Omega_k\cdot\big(\sum_{i=0}^{\ell-1}\frac{k}{n-i}\cdot(t_{i+1}-t_i)+\frac{k}{n-\ell}\cdot(t-t_\ell)\big)\Big) \tag{21}$$

The sequential analysis is very similar to that of gracefully degrading systems (Section 4.1) and hence omitted here.

4.3 Standby Redundant System (SRS)

In SRS, k-out-of-n cores are initially configured as active units, while the remaining $(n-k)$ cores are in spare mode. Upon detection of a permanent component failure, the system attempts to wake up a spare core and configure it as an active one. Note that, different from the strategy of PRS which aims to balance the age of all cores, in SRS only when some active cores fail, cold standbys might convert into active mode, which will lead to significant difference between cores in terms of age. For example, suppose the first core failure occurs when the system has been used for 4 years, after reconfiguration the system will be composed of $(n-1)$ 4-year old cores and a brand-new one.

Consider a core that starts its service life at time t^s. From t^s to its failure or the entire system's failure, its wearout rate is a constant Ω_k, because the quantity of active cores in the system is always k and this core is always one of them. As a result, its reliability only depends on its service time up to t while is independent of the failures in the systems, given by

$$R^{SRS}(t,t^s) = \mathcal{R}(\theta\cdot\Omega_k\cdot(t-t^s)) \tag{22}$$

To evaluate $MTTF^{SRS,sys}$, it is necessary to compute the probability that all surviving cores after ℓ failures are still operational at time t. It can be expressed by considering u_i (the quantity of surviving cores starting their service life from time t_i), i.e., $\prod_{i=0}^{\ell}\big(R^{SRS}(t,t_i)\big)^{u_i}$. Note that, u_i is function of past failure history h. As this event has a condition that h occurs, we can express $P(h)$ the probability of history h according to [9]. Let $\mathcal{H}$ be the set of all possible histories. According to the theorem of total probability, the unconditional reliability is therefore

$$R^{GDS,sys}(t,\ell) = \int\cdots\int_{\mathbf{D}} \left(R^{GDS}(t|\ell)\right)^{n-\ell} \prod_{i=0}^{\ell-1} \left((n-i)\cdot f^{GDS}(t_{i+1}|i)\right)\mathrm{d}t_1\cdots\mathrm{d}t_\ell \quad (17)$$

$$MTTF^{GDS,sys} = \mathrm{E}[\text{service life of gracefully degrading system}] = \int_0^\infty R^{GDS,sys}(t)\mathrm{d}t \quad (19)$$

$$R^{SRS,sys}(t,\ell) = \sum_{h\in\mathcal{H}} \int\cdots\int_{\mathbf{D}} \prod_{i=0}^{\ell} \left(R^{SRS}(t,t_i)\right)^{u_i(h)} \cdot P(h)\mathrm{d}t_1\cdots\mathrm{d}t_\ell \quad (23)$$

whose domain **D** is as same as that of Eq. (17). Similar to the analysis of gracefully degrading system, the expected service life is given by

$$MTTF^{SRS,sys} = \int_0^\infty R^{SRS,sys}(t)\mathrm{d}t = \int_0^\infty \sum_{\ell=0}^{n-k} R^{SRS,sys}(t,\ell)\mathrm{d}t \quad (24)$$

4.4 Extension to Heterogeneous System

Up to now, we have shown how to model the lifetime reliability of manycore systems with various redundancy configurations, wherein we regard the entire manycore processor as a k-out-of-n: G system. In practice, some designs may consist of more than one type of processor cores [8]: main processors and co-processors.This event can be modeled by simply extending our model presented in the previous sections. Assuming the failures within the two subsystem are independent, the lifetime reliability of the entire system comes down to the probability that both subsystems are operational.

Generally, consider a system that can be divided into q subsystems, in which subsystem i ($1 \le i \le q$) contains n_i identical components initially and functions if no less than k_i are operational. Each subsystem can has its own redundancy configuration scheme, referred as CFG_i in the superscript. Because of their different functionalities, we assume cores from different subsystems do not share workloads. The lifetime reliability of subsystem i at time t can then be obtained by substituting its parameters n_i and k_i into the models presented in Section 4, denoted as $R^{CFG_i,sys}(t)$. Recall that the functioning of all subsystems is essential for the entire system to operate properly. The expected service life of the entire system is hence given by

$$MTTF^{HS,sys} = \int_0^\infty \prod_{i=1}^{q} R^{CFG_i,sys}(t)\mathrm{d}t \quad (25)$$

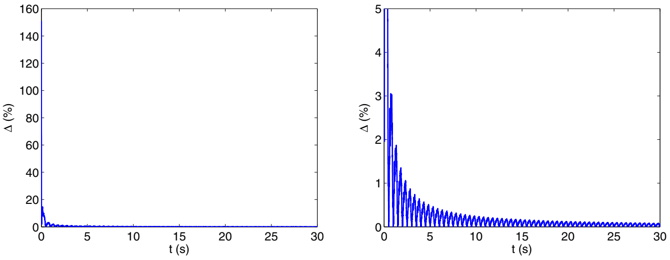
5. EXPERIMENTAL METHODOLOGY

To compare manycore systems with different redundancy configurations, we conduct extensive experiments on a 36-core processor (i.e., $n = 36$) with various workloads, with the number of active cores k ranging from 32 to 36. We implement a discrete event simulator to perform task allocation and scheduling for an application flow composed of 60,000 tasks in every experiment and we generate the associated power trace files for the entire system accordingly. Next, we take these files as the input of HotSpot tool [15] to acquire the temperature trace files. All temperature samples are collected to extract the temperature distribution (as in Fig. 1). We then compute the wearout rate Ω according to its definition, and finally obtain the lifetime reliability of manycore systems with various redundancy configurations, by computing multidimensional integral with Monte Carlo simulation.

5.1 Workload Description

As discussed earlier, workloads determine the temperature distribution of the processor. In our experiments, we generate a task flow for each workload, which is characterized by the task interarrival time distribution and the task service time distribution.

We assume the task interarrival time is with an exponential distribution with rate λ. Assuming that all the given m active cores equally share the workload, the task interarrival time of a core is $\frac{\lambda}{m}$. The task service time is modeled as exponential distribution and bimodal hyperexponential distribution. The exponential distribution is widely-used in the literature, while the bimodal hyperexponential distribution is regarded as the most probable distributions for modeling processor service time [19]. Exponential distribution has the expected service rate μ. Bimodal hyperexponential distribution is composed of two exponential distributions with mean $\frac{1}{\mu_1}$ and $\frac{1}{\mu_2}$ respectively, where $\frac{1}{\mu_1} = \frac{1}{\mu} + \frac{1}{\mu}\sqrt{\frac{(C_x^2-1)\alpha}{2(1-\alpha)}}$ and $\frac{1}{\mu_2} = \frac{1}{\mu} - \frac{1}{\mu}\sqrt{\frac{(C_x^2-1)(1-\alpha)}{2\alpha}}$. We set $\alpha = 0.95$, $C_x = 3.0$ [21]. For both distributions, the system load is defined as $\rho^{sys} = \frac{\lambda}{\mu}$. Consequently, each active core's load is $\rho = \frac{\lambda}{m\mu}$.



(a) Original Scale (b) Enlarged Scale

Figure 2: The Effectiveness of Wearout Rate Approximation.

5.2 Temperature Distribution Extraction

In our experiments, the die size of each core is set to be $5.76mm^2$ ($2.4mm \times 2.4mm$). Depending on its current workload, an active core can be in one of two states: *Run* and *Idle*. To represent the uneven power densities in the processing unit, a core contains a small block (e.g., Execution Unit) with higher power density, whose size is $0.5mm \times 0.5mm$. The power density values of this hotspot block and other parts in *Run* state are $5.0W/mm^2$ and $0.5W/mm^2$, while that in *Idle* state are both $0.16W/mm^2$. These system parameters are set according to state-of-the-art processors (e.g., IBM PowerPC 750CL [12]). The standbys are assumed to be in *Shut Down* state, whose power consumption is $\sim 0W$.

5.3 Reliability Factors

We use Weibull distribution, a well-accepted lifetime distribution for modeling hard errors of IC product [2], as the reliability function used in our system, $\mathcal{R}(t) = e^{-(\frac{t}{\theta})^\beta}$. shape parameter $\beta = 2.5$. Although the proposed approach is applicable for any failure mechanisms or their combinations, due to the lack of public empirical data on the relative weights of different failure mechanisms on real circuits, we analyze the electromigration failure in our experiments, whose models is presented in [7].

To compare the systems' lifetimes in various configurations, we normalize them to a certain scenario, wherein all cores of a 36-core system without redundancy are in active mode and the system workload is 5.0 and its Ω is set to be 0.1. In other words, a core in such a system has expected service life of 10 years.

6. RESULTS AND DISCUSSIONS

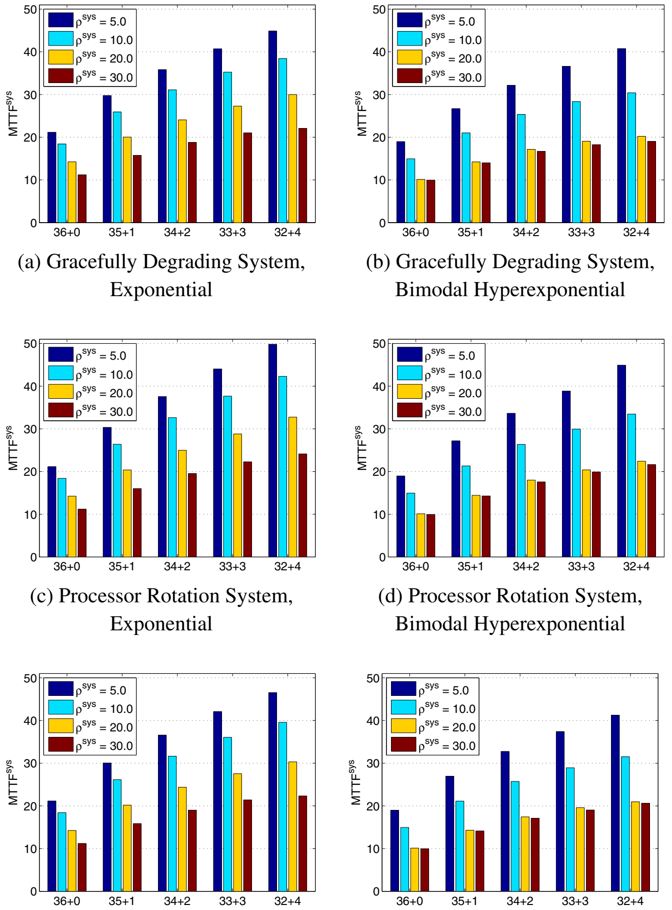
6.1 Wearout Rate Computation

We first demonstrate the effectiveness for one of the key concepts in this work, the wearout rate Ω computation, with experiments. On one hand, we trace the temperature variation of a core for 15,000 steps after the system has been warmed up, each corresponding to $3.33\mu s$, from which the temperature distribution $\nu(T)$ is extracted. The wearout rate Ω is then computed according to Eq. (13). From Eq. (12), the component reliability can be expressed as a function of $t \cdot \Omega$ (refer to as T_Ω hereafter). We compute T_Ω for 30 seconds. On the other hand, the temperature variation of the same core is traced for 30 seconds (around 9×10^6 steps) for comparison. We use T_{trace} to represent the summation of $\frac{\Delta t}{\theta_s(T,V,f)}$ up to time t, where $\Delta t = 3.33\mu s$. The difference between T_Ω and T_{trace} versus time t is shown in Fig. 2.

As can be seen from this figure, only in the first $4.832s$, the difference between the approximated Ω value and the actual value is larger than 0.5%. After that, T_Ω becomes very close to T_{trace}. Since the service life of processors is typically in the range of years, the estimation error by the proposed approach is negligible.

6.2 Comparison on Lifetime Reliability

Fig. 3 shows one of the most important metrics reflecting system lifetime, mean time to failure, under various redundancy configurations and workloads. With the increase of redundant cores, it is expected to have system lifetime extension and all the subfigures show this trend. Also, a closer observation of these figures show that the lifetime growth rate becomes smaller when more cores are configured as redundancy, i.e., the sojourn time of i-Failure state is larger than that of $(i+1)$-Failure state (see Fig. 4). This is mainly due to the increasing failure rate of IC products. Consider three PRS systems with 0, 2, and 4 redundant cores as an example (the three middle bars in Fig. 4). From 36+0 to 34+2 systems,



(a) Gracefully Degrading System, Exponential

(b) Gracefully Degrading System, Bimodal Hyperexponential

(c) Processor Rotation System, Exponential

(d) Processor Rotation System, Bimodal Hyperexponential

(e) Standby Redundant System, Exponential

(f) Standby Redundant System, Bimodal Hyperexponential

Figure 3: Comparison of Different Redundancy Configurations under Various Workloads.

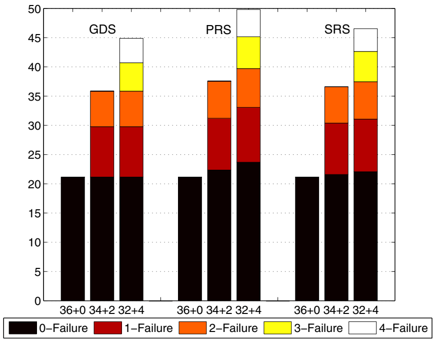
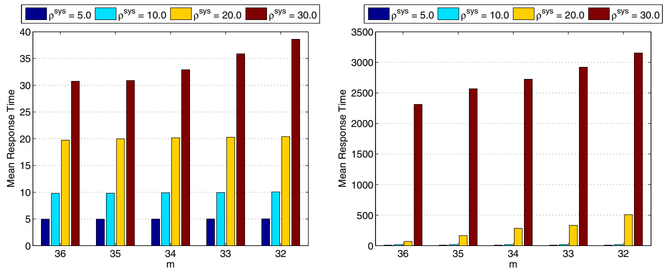


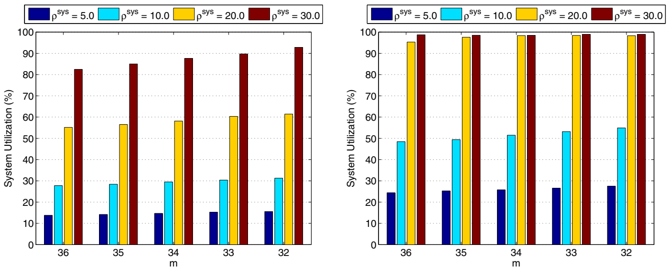
Figure 4: Detailed Sojourn Time in Various States.



(a) Exponential

(b) Bimodal Hyperexponential

Figure 5: Mean Response Time under Various Workloads.



(a) Exponential

(b) Bimodal Hyperexponential

Figure 6: System Utilization under Various Workloads.

the sojourn time in 0-Failure state rises from 21.16 to 22.35, increased by 1.19, while the additional 1-Failure and 2-Failure states for the 34+2 system provide 8.87 and 6.35 extra service life, respectively. As a result, the overall lifetime extension is 16.41. If we further increase the number of redundant cores by two, the lifetime extension is 12.25, less than 16.41. From the above, improving the lifetime reliability of manycore processors by increasing the value of k gradually diminishes and it may not quite beneficial to set it as a very large value.

From Fig. 3 and Fig. 4, we can also see that PRS provides longer service life than the other two configurations under the same workloads. On one hand, when compared to standby systems, processor cores in PRS have a more balanced workload. That is, in SRS, some cores are set as cold standbys initially and convert to active mode only when some active cores fail. Thus, even if the workload is evenly distributed among all active cores, after some replacements the system is composed of many old components and a few new ones. Since the aged cores have already had high failure rate, although there are some new cores, the lifetime of the entire system cannot be extended much. From this aspect, a lot of potential computation capabilities of standbys are wasted. This problem can be avoided by using PRS configuration, which aims to balance the aging effect among all cores. On the other hand, when compared to gracefully degrading systems, processor cores in PRS alternate between active and standby modes while all cores in GDS keep aging in its lifetime. Although PRS can result in slightly heavier workload on every single core than GDS, the extra aging effects because of this issue is quite small when k is much smaller than n.

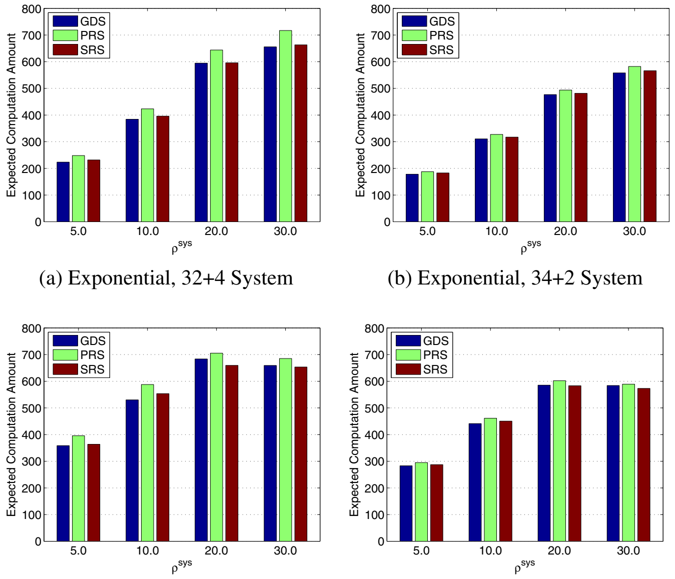
6.3 Comparison on Performance

The various redundancy configurations also result in different performances for the manycore processors. Two widely-used metrics, *mean response time* and *system utilization*, defined as the expected time from a task's arrival until its completeness and the average percentage of cores under-usage over time, respectively, are used to evaluate the performances of the manycore systems. The results are achieved by using the same discrete-event simulator.

Fig. 5 shows the mean task response time versus the number of active cores in the system under various workloads. Consider exponential service time first (Fig. 5(a)). When the workload is not high ($\rho^{sys} \leq 20.0$), the mean response time slightly increases with the decline number of active cores. In addition, this value roughly doubles as the workload becomes twice larger. For instance, when all 36 cores serve as active ones, the mean response times corresponds to ρ^{sys} =5.0, 10.0 and 20.0 are 4.96, 9.79, and 19.73, respectively. When the workload is high (i.e., $\rho^{sys} = 30.0$), the mean response time is still roughly proportional to workloads, but a few less active cores lead to a noticeable increment of response time. For bimodal hyperexponential service time (Fig. 5(b)), while the systems with $\rho^{sys} = 30.0$ have similar lifetime with $\rho^{sys} = 20.0$ (see Fig. 3), their mean response time is significantly larger (ranging between $6.5 - 33.4\times$). We attribute this phenomenon to the close-to-saturated system workload under such circumstances. In other words, with the parameters setup in our experiments, when the workload of systems with bimodal hyperexponential distribution becomes larger than 20.0, almost all active cores always have tasks to perform, thus leading to the dramatic increase of the mean response time of tasks.

When it comes to the performances of various redundancy configurations, a GDS system sequentially has 36, 35, $\cdots$ active cores in its lifetime and therefore experiences gracefully degrading performance from the users' point of view. Other configurations, by contrast, are with the same performance over its lifetime. For example, a PRS/SRS system that has 32 active cores at any time always provides mean task response time 38.60, given exponential service time and $\rho^{sys} = 30.0$. A 32+4 GDS system, however, is able to provide better performance in the first several years (its mean response time is 30.75), and then gradually increase to 30.89, 32.90, 35.89, and finally 38.60.

Fig. 6 shows the system utilization in various cases. For exponential service time, the system utilization is almost proportional to their system workload with the number of active cores. Under the fixed workloads, we can also observe slightly higher system utilization for systems with less active cores. When the workload becomes sufficiently heavy with hyperexponential distributed service time (when $\rho^{sys} \geq 20.0$), the system utilization increases very little with the increment of workloads. This can well explain the mean response time shown in Fig. 5(b).



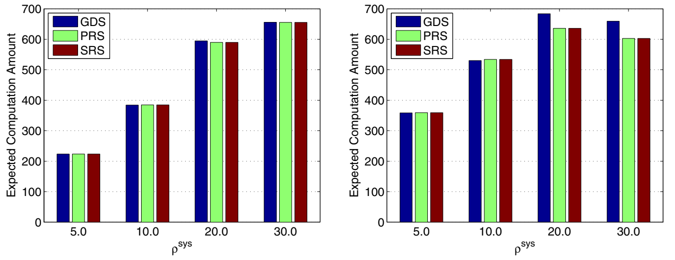
(a) Exponential, 32+4 System

(b) Exponential, 34+2 System

(c) Bimodal Hyperexponential, 32+4 System

(d) Bimodal Hyperexponential, 34+2 System

Figure 7: Expected Computation Amount Before System Failure.



(a) Exponential

(b) Bimodal Hyperexponential

Figure 8: Comparison of Three Redundancy Configurations in Expected Computation Amount with the Same Service Time.

6.4 Comparison on Expected Computation Amount

In this subsection, we combine the performance and lifetime reliability into a unified metric, namely *expected computation amount*, which reflects the amount of computation performed by a system before its failure. The results for 32+4 and 34+2 systems are shown in Fig. 7. An interesting phenomenon can be observed from these figures. That is, as the system workload becomes heavier, in spite of significant decline in the system lifetime (see Fig. 3), in most cases the total computation amount of the system increases. This is mainly because, although the system with light workload has relatively lower temperature when compared with that with heavy load, the induced difference in lifetime is less than the difference in system utilization. In particular, consider GDS shown in Fig. 7(a) as an example. The sojourn time in 0-Failure states for $\rho^{sys} = 5.0$ and 10.0 is 21.16 and 18.42, respectively, while the system utilization of two cases is 13.80% and 27.76%. Therefore, the expected computation amount ratio in this state is around 1.75, and we can observe similar trends in other states. From this aspect, longer service life does not mean more effective usage of the manycore processor. Moreover, we notice that when ρ^{sys} of the system with bimodal hyperexponential service time distribution increases from 20.0 to 30.0, the expected computation amount decreases (see Fig. 7(c)-(d)). The main reason lies in the fact that both cases nearly make full use of their resources and thus their computation amounts are mainly bounded by their service lives.

In some cases, we may not use computer systems until the end of their lifetimes. Hence, we are also interested in the computation amount of systems under such situations. In the following experiments, we set the minimum expected service life among GDS, PRS, and SRS computed by the proposed model as the actual system service life, and calculate the expected computation amount until that time point for the three redundancy configurations. The results for 32+4 systems are shown in Fig. 8. When the systems are not fully used (i.e., exponential service time with $\rho^{sys} = 5.0$, 10.0, 20.0, and 30.0 and bimodal hyperexponential with $\rho^{sys} = 5.0$ and 10.0), we can see that the total computation amounts with different configurations are the same. This is because, as the system is not fully utilized, it is always able to complete tasks within a very short time period (compared to the system's lifetime). Therefore, the computation amount equals to the task amount feeded to the system. If the system has sufficient high utilization (i.e., hyperexponential distributed service time with $\rho^{sys} = 20.0$ and 30.0 in Fig. 8(b)), GDS systems finish more jobs than the other two configurations. This is expected because GDS systems contain more active units and keep them busy, leading to greater computation amount. It is also worth to note that with the same workload distribution, PRS and SRS always have the same computation amount when considering the same service time. This is because in both configurations the number of active cores at any time remains the same (32, in our experiments).

7. CONCLUSION

In this paper, we propose a novel analytical model to characterize the lifetime reliability of manycore processors with various redundancy configurations. Our proposed model is able to capture the impact of temperature variations of processor cores and workloads. Our experiments compare the lifetimes and performances of gracefully degrading systems, processor rotation systems and standby redundant systems, under various workloads.

8. ACKNOWLEDGEMENTS

This work was supported in part by Hong Kong SAR Research Grants Council (RGC) under the General Research Fund CUHK417807, and CUHK418708, in part by National Science Foundation of China (NSFC) under grant No 60876029, and in part by a grant N_CUHK417/08 from the NSFC/RGC Joint Research Scheme.

9. REFERENCES

[1] Methods for calculating failure rates in units of fits (jesd85). *JEDEC Publication*, 2001.

[2] Failure mechanisms and models for semiconductor devices (jep122c). *JEDEC Publication*, 2003.

[3] J. R. Black. Electromigration - A Brief Survey and Some Recent Results. *IEEE Transactions on Electron Devices*, 16(4):338–347, April 1967.

[4] S. Borkar. Thousand Core Chips - A Technology Perspective. In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pages 746–749, 2007.

[5] A. Coskun, T. Rosing, K. Mihic, G. D. Micheli, and Y. L. Lebici. Analysis and optimization of mpsoc reliability. *Journal of Low Power Electronics*, 15(2):159–172, February 2006.

[6] D. Geer. Chip Makers Turn to Multicore Processors. *IEEE Computer*, 38(5):11–13, May 2005.

[7] A. K. Goel. *High-Speed VLSI Interconnections*. IEEE Press, 2nd edition, 2007.

[8] M. D. Hill and M. R. Marty. Amdahl's Law in the Multicore Era. *IEEE Computer*, 41(7):33–38, July 2008.

[9] L. Huang and Q. Xu. On Modeling the Lifetime Reliability of Homogeneous Manycore Systems. In *Proceedings Pacific Rim International Symposium on Dependable Computing*, pages 87–94, 2008.

[10] L. Huang, F. Yuan and Q. Xu. Lifetime Reliability-Aware Task Allocation and Scheduling for MPSoC Platforms. In *Proc. IEEE/ACM Design, Automation, and Test in Europe (DATE)*, pages 51–56, 2009.

[11] L. Huang and Q. Xu. Lifetime Reliability for Load-Sharing Redundant Systems with Arbitrary Failure Distributions. *IEEE Transactions on Reliability*, to appear.

[12] IBM. IBM PowerPC 750CL Microprocessor Revision Level DD2.x. http://www-01.ibm.com/chips/techlib/techlib.nsf/techdocs/2F33B5691BBB8769872571D10065F7D5/$file/750cldd2x_ds_v2.4_pub_29May2007.pdf.

[13] J. Shin, V. Zyuban, P. Bose, and T. M. Pinkston. A Proactive Wearout Recovery Approach for Exploiting Microarchitectural Redundancy to Extend Cache SRAM Lifetime. In *Proceedings IEEE/ACM International Symposium on Computer Architecture (ISCA)*, pages 353–362, 2008.

[14] J. Shin, V. Zyuban, Z. Hu, J. Rivers, and P. Bose. A Framework for Architecture-Level Lifetime Reliability Modeling. In *Proceedings IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 534–53, 2007.

[15] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-Aware Microarchitecture. In *Proceedings IEEE/ACM International Symposium on Computer Architecture (ISCA)*, pages 2–13, 2003.

[16] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. The Case for Lifetime Reliability-Aware Microprocessors. In *Proceedings IEEE/ACM International Symposium on Computer Architecture (ISCA)*, pages 276–287, 2004.

[17] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. Exploiting Structural Duplications for Lifetime Reliability Enhancement. In *Proceedings IEEE/ACM International Symposium on Computer Architecture (ISCA)*, pages 520–531, 2005.

[18] A. Tiwari and J. Torrellas. Facelift: Hiding and Slowing Down Aging in Multicores. In *Proceedings International Symposium on Microarchitecture (MICRO)*, pages 129–140, 2008.

[19] K. S. Trivedi. In *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley & Sons, second edition, 2002.

[20] M. Xie, Y.-S. Dai, and K.-L. Poh. In *Computing Systems Reliability: Models and Analysis*. Kluwer Academic Publishers, 2004.

[21] B. S. Yoo and C. R. Das. A fast and efficient processor allocation scheme for mesh-connected multicomputers. *IEEE Transactions on Computers*, 51(1):46–60, January 2002.

[22] S. Zafar, A. Kumar, E. Gusev, and E. Cartier. Threshold Voltage Instabilities in High-κ Gate Dielectric Stacks. *IEEE Transactions on Device and Materials Reliability*, 5(1):45–64, March 2005.