The Effects of Energy Management on Reliability in Real-Time Embedded Systems

Dakai Zhu, Rami Melhem and Daniel Mossé
Computer Science Department
University of Pittsburgh
Pittsburgh, PA 15260
{zdk, melhem, mosse}@cs.pitt.edu

Abstract— **The slack time in real-time systems can be used by recovery schemes to increase system reliability as well as by frequency and voltage scaling techniques to save energy. Moreover, the rate of transient faults (i.e., soft errors caused, for example, by cosmic ray radiations) also depends on system operating frequency and supply voltage. Thus, there is an interesting trade-off between system reliability and energy consumption. This paper first investigates the effects of frequency and voltage scaling on the fault rate and proposes two fault rate models based on previously published data. Then, the effects of energy management on reliability is studied. Our analysis results show that, energy management through frequency and voltage scaling could dramatically reduce system reliability, and ignoring the effects of energy management on the fault rate is too optimistic and may lead to unsatisfied system reliability.**

I. Introduction

For autonomous critical real-time embedded applications, such as satellite and surveillance systems, both high level of reliability and low energy consumption are desired. Fault tolerance through redundancy [13], [18] as well as energy management through frequency and voltage scaling [17], [26] have been well studied in the context of real-time systems. However, there are relatively less research addressing the combination of fault tolerance and energy management.

Recovery through rollback execution is a cost effective technique to tolerate transient faults and increase system reliability by exploring the slack time in real-time systems [18]. With technology advancement, in addition to being used for temporal redundancy, slack time can also be used by frequency and voltage scaling techniques to save energy by reducing system operating frequency and supply voltage [17], [26]. When more slack is dedicated to frequency and voltage scaling to save more energy, less slack is left for fault tolerance, which reduces the number of possible recoveries and thus reduces system reliability. Moreover, the rate of transient faults (i.e., soft errors caused, for example, by cosmic ray radiations) also depends on system operating frequency and supply voltage [1], [20], [23], [28], which makes the trade-off between system reliability and energy consumption more interesting.

Closely Related Work: For a set of independent periodic tasks, using the primary/backup recovery model, Unsal *et al.* proposed an energy-aware software-based fault tolerance scheme, which postpones as much as possible the execution of backup tasks to minimize the overlap of primary and backup execution and thus to minimize energy consumption [25]. For a single application to tolerate one fault, the optimal number of checkpoints, evenly or unevenly distributed, to achieve minimal energy consumption was explored for Duplex systems in [15]. Elnozahy *et al.* proposed an *Optimistic TMR* (OTMR) scheme to reduce the energy consumption for traditional TMR systems by allowing one processing unit to slow down provided that it can catch up and finish the computation before the application deadline [6]. The optimal frequency settings for OTMR was further explored in [27].

Previous research either focused on tolerating fixed number of faults [6], [15] or assumed constant fault rate [27] when applying frequency and voltage scaling for energy savings. To the best of our knowledge, this work is the first attempt to consider fault rate changes when exploring the trade-off between reliability and energy consumption in real-time embedded systems. We first propose two fault rate models related to frequency and voltage scaling based on previously published data. Then we study the effects of energy management on reliability.

The rest of this paper is organized as follows. The application, energy and fault models are presented in Section II. The effects of frequency and voltage scaling on fault rates are discussed and two fault rate models are proposed in Section III. Section IV analyzes the trade-off between reliability and energy consumption. The analytical results are presented and discussed in Section V and Section VI concludes the paper.

II. Models and Problem Description

A. Application Model

We consider a frame-based real-time application that is generally characterized by a *worst case execution time* (WCET), L, and a *deadline*, D, which is also the frame size. The application is executed repeatedly within every frame and we will focus on the execution of the application within one frame due to periodicity. For processor with variable frequency, such as Intel Pentium M processors [4], an application needs more time when running at lower frequency. For clarification, we assume that the WCET of an application, L, corresponds to executing the application at the maximum frequency f_{max}. Moreover, we use normalized frequency and assume $f_{max} = 1$. As usual in real-time systems, we assume that $L \leq D$ and the system load is defined as $\sigma = \frac{L}{D}$. Although we focus on frame based real-time applications, the framework proposed in this paper can be easily extended to periodic tasks, as discussed in [15].

Because of memory accesses caused by cache misses, the speedup of a system may not be linear with increased frequency [22]. However, for general applications with a reasonable size cache, the linear relation between speedup and frequency has very small variations [15]. For simplicity, we assume that the execution time of an application is linearly related to the frequency. That is, when frequency is reduced by half, the execution time doubles[1].

B. Power Model and Energy Management

In embedded systems, the power is consumed mainly by the processor, memory and underlying circuits. While the power consumption is dominated by dynamic power dissipation, which is quadratically related to supply voltage and linearly related

[1]Notice that this is a conservative model. The execution time of an application at reduced frequencies will be less than the modeled time considering that memory latency is independent of the processor frequency.

0-7803-8702-3/04/$20.00 ©2004 IEEE.

to frequency [2], the static leakage power cannot be ignored, especially with increased levels of integration [24]. Thus, the power consumption P in an embedded system can be modeled as [27]:

$$P = P_s + \hbar(P_{ind} + P_d) \quad (1)$$

$$P_d = C_{ef} V^2 f \quad (2)$$

where P_s is the *sleep power* (e.g., the power used to maintain basic circuits, keep the clock running etc., which can only be removed by turning off the system), P_{ind} is the *frequency-independent active power* (that consists of the components of memory and processor power that can be efficiently removed by putting systems to sleep and is independent of system supply voltage and frequency [4], [19]) and P_d is the *frequency-dependent active power* (that includes processor dynamic power and any power that depends on system supply voltage and frequency [2], [24]). $\hbar$ equals 0 if the system is in sleep state and $\hbar$ equals 1 otherwise[2]. C_{ef} is the switch capacitance, V is the supply voltage and f is the frequency.

The system can be put into sleep state after finishing the computation. However, considering the huge overhead of turning on/off a system [5], we assume that the system is always on. That is, the sleep power P_s is always consumed and is not manageable. Different P_s will not affect the absolute energy savings and, for simplicity, we will ignore the sleep power P_s (i.e., $P_s = 0$) and concentrate on the frequency-independent active power P_{ind} and frequency-dependent active power P_d in our analysis.

Frequency scaling saves P_d by reducing frequency without changing supply voltage [16]. Suppose the fixed supply voltage is V_{max} (the maximum supply voltage corresponding to f_{max}). At frequency f, the system power dissipation is $P = P_{ind} + C_{ef} V_{max}^2 f$ and the application needs $\frac{L}{f}$ time units. Since energy is defined as the integral of power over time, the energy consumption E to execute the application at frequency f and voltage V_{max} is (recall that f is a normalized frequency and $f_{max} = 1$):

$$E = (P_{ind} + C_{ef} V_{max}^2 f)\frac{L}{f} = P_{ind}\frac{L}{f} + C_{ef} V_{max}^2 L \quad (3)$$

Differentiating Equation 3 with respect to f, we get $\frac{\partial E}{\partial f} = 0$ if $P_{ind} = 0$. That is, the energy consumption to execute an application is independent of f if there is no frequency-independent active power. However, when there is frequency-independent active power ($P_{ind} > 0$), we have $\frac{\partial E}{\partial f} < 0$, and hence frequency scaling consumes more energy to execute an application at lower frequencies. Intuitively, when running at lower frequencies, the application consumes the same amount of frequency-dependent energy because of linearly decreased frequency-dependent active power and linearly increased execution time, but it consumes more frequency-independent energy with longer execution time.

Voltage scaling reduces the supply voltage for lower frequencies to save energy [17]. Since the circuit delay is almost linearly related to $\frac{1}{V}$ [2], for systems to function correctly, the operating frequency needs to decrease linearly when supply voltage is reduced. Inversely, when an application runs at lower frequencies, the supply voltage can be reduced linearly. Similar to frequency, normalized voltages are used and the maximum supply voltage V_{max} is assumed to be 1. Thus, for frequency f, the corresponding supply voltage is $V = f \cdot V_{max} = f$. Therefore, the energy consumption E to execute the application at f and V is:

$$E = (P_{ind} + C_{ef} V^2 f)\frac{L}{f} = P_{ind}\frac{L}{f} + C_{ef} f^2 L \quad (4)$$

From Equation 4, we can see that the lower the frequency is, the lower the supply voltage can be and the less frequency-dependent energy is consumed. However, more time is needed by an application and more frequency-independent energy is consumed. Therefore, there is an energy efficient frequency f_{ee} to minimize the energy consumption [7], [27]. Differentiating Equation 4 with respect to f, E is minimized when $f_{ee} = \sqrt[3]{\frac{P_{ind}}{2C_{ef}}}$. Suppose that the lowest frequency in a system is f_{low}, we define the *minimum energy efficient* frequency as $f_{min} = \max\{f_{low}, f_{ee}\}$. That is, we may be forced to run at a frequency higher than f_{ee} to meet the application deadline or to comply with the lowest frequency limitation, but we should never run at a frequency below f_{ee}, since doing so increases energy consumption. For simplicity, it is assumed that frequency can be changed continuously[3] between f_{min}, the minimum energy efficient frequency, and f_{max}, the maximum frequency. So does the supply voltage between V_{min} (corresponding to f_{min}) and V_{max} (corresponding to f_{max}).

C. Fault Model and Problem Description

During the execution of an application, a fault may occur due to various reasons, such as hardware failures, software errors and the effect of cosmic ray radiations. Since *transient* and *intermittent* faults occur much more frequently than *permanent* faults [3], in this paper, we focus on transient and intermittent faults, especially the ones caused by cosmic ray radiations, and explore the temporal redundancy to tolerate them. We assume that a self-checking fault detection mechanism is used [18].

For a real-time application with WCET L and a deadline D, the amount of slack time is $D - L$. The slack can be used to either reduce the frequency and/or supply voltage for energy savings or to schedule recoveries for higher *performability*, which is defined as *the probability of finishing the application correctly within its deadline in the presence of faults* [12]. Furthermore, as discussed in Section III, the fault rate also depends on the frequency and supply voltage.

Intuitively, the lower the frequency and supply voltage are, the less energy is consumed; however, there is less slack time left for recovery and the performability may decrease. In this work, considering two different fault rate models related to frequency and voltage scaling, we analyze the trade-off between system performability and energy savings.

III. The Effect of Frequency and Voltage Scaling on Fault Rates

Transient faults caused by radiations in semiconductor circuits have been known and well studied since the late 1970s [28]. When high-energy particles strike a sensitive region in semiconductor devices, a dense track of electron-hole pairs are deposited. The charge may be collected by pn-junctions via drift

[2]For systems that cannot be put into a sleep state, $\hbar$ always equals 1. This paper considers only the systems that can be put to a sleep state.

[3]For discrete frequency levels, we can use two adjacent levels to emulate the execution at any frequency [10].

and diffusion mechanisms to form a current pulse and cause a logic error [11], or to accumulate and exceed the minimum charge (i.e., the *critical charge*) required to flip the value stored in a memory cell [9], [29].

Because it is relatively easy to detect errors in memory and large areas in microprocessor chips are dedicated to caches and registers, numerous work has examined the effect of cosmic ray radiations on memory circuits [9], [21], [29]. Although past research has shown that logic circuits are less susceptible to cosmic ray radiations than memory [14], a recent model predicts that, with technology advancement and reduced feature size, the fault rate in combinational logic circuits will be comparable to that of memory elements [23]. There are various factors that affect the fault rate, such as cosmic ray flux (i.e., number of particles per area), technology feature size, chip capacity, supply voltage and operating frequency, and thus, modeling the fault rate is extremely hard [21], [23], [29].

In this work, we focus on the effects of frequency and voltage scaling on fault rate changes. We assume that the radiation induced faults follow a Poisson distribution with an average fault rate λ being determined by system supply voltage and frequency. For simplicity, no variation of cosmic ray flux or other factors are considered. That is, for a given supply voltage and frequency (e.g., V_{max} and f_{max}), the average fault rate (e.g., λ_0) is fixed. Hence, for systems running at frequency f ($f_{min} \leq f \leq f_{max}$) and voltage V ($V_{min} \leq V \leq V_{max}$), the general model for the average fault rate can be expressed as:

$$\lambda(f, V) = \lambda_0 \cdot g(f, V) \tag{5}$$

where λ_0 is the average fault rate corresponding to V_{max} and f_{max}. That is, $g(f_{max}, V_{max}) = 1$. In what follows, we consider two different models for the fault rate based on previously published data: the linearly decreasing fault rate model for frequency scaling and the exponentially increasing fault rate model for voltage scaling. However, the framework for exploring the trade-off between system reliability and energy consumption proposed in this paper is independent of fault rate models.

A. Linear Model for Frequency Scaling (with Fixed Voltage)

When the supply voltage is fixed, the fault rate in combinational logic circuits [1], [8] as well as memory [23] was shown to *decrease* linearly when frequency is reduced. This comes from the safety margin in clock cycles becoming relatively larger when the frequency decreases [1], [23]. Thus, for frequency scaling (with fixed supply voltage), the average fault rate at frequency f can be modeled as:

$$\lambda(f, V) = \lambda(f) = \lambda_0 \cdot f^b \tag{6}$$

where b (> 0) is a constant. When $b = 1$, the fault rate is linearly increasing with the frequency.

B. Exponential Model for Voltage Scaling

For different technologies that have different supply voltages, Seifert *et al.* examined the fault rate in the family of Alpha processors due to α particle effects using both simulations and experiments [20]. Their results showed that fault rates in these processors (including logic core and cache) increases exponentially when supply voltage decreases. The same observation has been shown in [29] for memory. The reason is that, with reduced supply voltage, the critical charge becomes smaller which results in exponentially increased fault rate [9], [23]. Moreover, there are many more lower energy particles than higher energy particles (e.g., one order of magnitude less in energy corresponding to 100 times more in the number of particles) [28]. With smaller critical charge, lower energy particles could cause an error.

As discussed in Section II, voltage scaling reduces supply voltage for lower frequencies [17]. We use the conclusions in [9], [20], [23], [28], [29] to formulate the effects of voltage scaling on fault rates. At the lowest frequency f_{min} and supply voltage V_{min}, the average fault rate is assumed to be $\lambda_{max} = \lambda_0 10^d$, where d (> 0) is a constant. When a system runs at frequency f and corresponding voltage $V = f \cdot V_{max} = f$, the average fault rate can be expressed as (recall that normalized frequency and voltage are used):

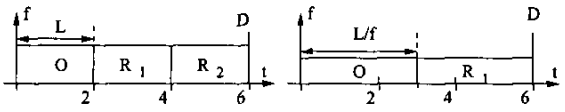
$$\lambda(f, V) = \lambda(f) = \lambda_0 10^{\frac{d(1-f)}{1-f_{min}}} \tag{7}$$

That is, reducing the supply voltage for lower frequency results in exponentially increased fault rates and larger d indicates that the fault rate is more sensitive to voltage scaling.

IV. Trade-off between Reliability and Energy Consumption

When a transient fault is detected during the execution of an application, it can be tolerated by re-executing the application. The overhead of reloading the application for re-execution is assumed to be incorporated into the application's WCET L. At frequency f ($\leq f_{max}$), the number of re-executions that can be scheduled within the application deadline D is:

$$k_f = \left\lfloor \frac{Df}{L} \right\rfloor - 1 \tag{8}$$



a. f_{max} = 1 with 2 recoveries b. f=2/3 with 1 recoveries

Fig. 1. For an application with $L = 2$ and $D = 6$, there are two recoveries when running at $f_{max} = 1$ and one recovery at $f = \frac{2}{3}$.

That is, lower frequencies lead to fewer possible recoveries. For example, as illustrated in Figure 1, if an application has $L = 2$ and $D = 6$, two recoveries can be scheduled within the application deadline at the maximum frequency f_{max} (Figure 1a). However, when the frequency is reduced to $f = \frac{2}{3}$, only one recovery is possible (Figure 1b). In the figures, the X-axis represents time, the Y-axis represents operating frequency, and the area of a box defines the amount of work needed to execute the application.

In this work, we assume that an application and its recoveries run at the same frequency. Considering the probability of the recoveries being invoked (e.g., one recovery is invoked only if an application and all the previous recoveries fail), it may be more energy efficient to run the original execution with a lower frequency and the recoveries with higher frequencies. We will explore this point in our future work.

A. The Performability

At frequency f and supply voltage V, the average fault rate is $\lambda(f, V)$. Assuming that faults follow Poisson distribution, the probability of having at least one fault during one running of the application is:

$$\rho_f = 1 - e^{-\lambda(f,V)\frac{L}{f}} \tag{9}$$

Thus, when there are k_f recoveries, the performability is:

$$R_f = 1 - \rho_f^{k_f+1} = 1 - \left(1 - e^{-\lambda(f,V)\frac{L}{f}}\right)^{\lfloor \frac{Df}{L} \rfloor} \tag{10}$$

where $\rho_f^{k_f+1}$ is the probability of having fault(s) during every execution (including the original execution and the k_f recovery executions).

For frequency scaling, where the supply voltage is fixed, the average fault rate is modeled as $\lambda(f, V) = \lambda(f) = \lambda_0 f^b$ ($b > 0$; see Section III). Thus, the probability of having fault(s) during one running of the application is $\rho_f = 1 - e^{-\lambda_0 f^b \frac{L}{f}} = 1 - e^{-\lambda_0 L f^{b-1}}$. When $b \leq 1$, we have $\frac{\partial \rho_f}{\partial f} \leq 0$, that is, ρ_f increases when frequency f decreases. Notice that, lower frequencies result in fewer number of recoveries (i.e., smaller k_f; see Equation 8). Hence, system performability R_f decreases when frequency f decreases (see Equation 10).

When $b > 1$, we have $\frac{\partial \rho_f}{\partial f} > 0$ and ρ_f decreases when f decreases. Because of integer limitation on the number of recoveries k_f (the floor in Equation 8), different frequencies may result in the same number of recoveries. For those frequencies, system performability R_f is higher for lower frequencies due to decreased ρ_f. However, when lower frequency f leads to fewer recoveries, system performability R_f decreases dramatically (see the analysis results in Section V).

With voltage scaling, where the supply voltage is reduced for lower frequencies to save more energy, we have $\lambda(f, V) = \lambda(f) = \lambda_0 10^{\frac{d(1-f)}{1-f_{min}}}$ ($d > 0$; see Section III). From Equation 9, we have $\frac{\partial \rho_f}{\partial f} < 0$. That is, ρ_f increases when frequency f decreases. Recall that lower frequencies result in fewer number of recoveries (see Equation 8). Therefore, from Equation 10, the system performability R_f decreases when supply voltage is reduced for lower frequencies.

B. The Expected Energy Consumption

From Section II, the energy consumption to execute the application once at frequency f and voltage V is $E_f = (P_{ind} + C_{ef} V^2 f)\frac{L}{f}$. Notice that, the original execution needs to be performed under all circumstances. If the original execution fails, the first recovery is executed with the probability of ρ_f, which is the probability of having fault(s) during the original execution. Similarly, the i^{th} ($i \leq k_f$) recovery will be executed with probability ρ_f^i, which is the probability of having fault(s) during the original execution as well as every previous recovery execution. Thus, the *expected energy consumption* is:

$$EE_f = E_f \sum_{i=0}^{k_f} \rho_f^i = E_f \frac{1 - \rho_f^{k_f+1}}{1 - \rho_f} \tag{11}$$

For frequency scaling, we have $V = V_{max}$ and $E_f = (P_{ind} + C_{ef} V_{max}^2 f)\frac{L}{f} = P_{ind}\frac{L}{f} + C_{ef} V_{max}^2 L$. Thus, E_f is a constant when frequency-independent active power $P_{ind} = 0$; when $P_{ind} > 0$, E_f increases when the frequency f to execute the application decreases. Notice that, in general, $1 - \rho_f^{k_f+1} \approx 1$ and $P_{ind} > 0$. From Equation 11, we see that the expected energy consumption EE_f increases when f decreases due to increased E_f and/or increased ρ_f (see Section IV-A).

Under voltage scaling, if frequency-independent active power $P_{ind} = 0$, the energy consumption to execute the application once at frequency f and corresponding voltage V is $E_f = C_{ef} V^2 f \frac{L}{f} = C_{ef} f^2 L$. That is, lower frequencies and voltages are more energy efficient for executing the applications once. For the expected energy consumption EE_f, we have $EE_f < \lfloor \frac{Df}{L} \rfloor E_f$, where $\lfloor \frac{Df}{L} \rfloor E_f$ is the pessimistic estimation of the expected energy consumption assuming that all recoveries are executed. Notice that, when the application runs at f_{max}, we have $E_{f_{max}} = C_{ef} f_{max}^2 L = C_{ef} L$ and $E_{f_{max}} < EE_{f_{max}}$. If $\frac{Df}{L} E_f \leq E_{f_{max}}$ (i.e., $\frac{f^3}{\sigma} \leq 1$), we have $EE_f < EE_{f_{max}}$. That is, when frequency f satisfies $\sigma \leq f \leq \sqrt[3]{\sigma}$, regardless of performability, the expected energy consumption running the application at f and corresponding voltage V is always less than that of f_{max}. When $P_{ind} > 0$ or $f > \sqrt[3]{\sigma}$, the expected energy consumption may increase for lower frequencies and voltages when the probability of executing the recoveries is rather high due to fast increased fault rates. This interesting case is further illustrated in Section V (Figure 3b).

V. Analysis Results and Discussion

In this section, we will present some analytical results that show the significant effects of energy management on performability by considering fault rate changes caused by frequency and voltage scaling. First, let us determine some system parameters. As discussed in Section II, we use normalized frequency and voltage with $f_{max} = 1$ and $V_{max} = 1$. Furthermore, we assume that the maximum frequency-dependent active power is $P_d^{max} = C_{ef} V_{max}^2 f_{max} = 1$ and use normalized values for the frequency-independent active power P_{ind}. Considering that the Intel Pentium M processor consumes $25W$ peak power with sleep power around $1W$ [4] and that the RAMBUS memory chip consumes $300mW$ active power with nap power[4] of $30mW$ [19], we use the values of $0.0, 0.2$ and 0.4 for P_{ind}.

For the rate of radiation induced faults, the number of 1000 to 100,000 FITs (failure in time, in terms of errors per billion hours of use) per megabit as a reasonable fault rate range has been reported, which corresponds to 10^{-8} to 10^{-6} faults per second on each chip (assuming a 100 megabit chip) and 10^{-6} to 10^{-2} faults per second in a system [21], [23], [29]. In this work, we assume that $\lambda_0 = 10^{-6}$ at voltage V_{max} and frequency f_{max}. We vary the values of b (as $0.1, 1$ and 10, respectively) and d (as $0, 2, 4$ and 6 respectively) for different changes in fault rates due to frequency and voltage scaling.

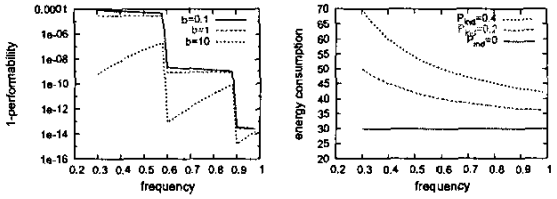
We consider an application that has a deadline $D = 100$ time units and the worst case execution time[5] $L = 30$ at the maximum frequency f_{max}.

[4]As indicated in [7], it is more energy efficient to put RAMBUS memory into nap state instead of sleep state because of the long latency to bring the chip back from sleep to active.

[5]For different workloads with other values of L (e.g., $L = 40$ and $L = 50$), similar results are obtained. Instead of presenting load variations, the focus of our analysis is on different fault rates caused by frequency and voltage scaling.

A. Frequency Scaling with Linearly Decreased Fault Rates

From Section III, with fixed supply voltage $V_{max} = 1$, the fault rate decreases when frequency decreases, and we have $\lambda(f, V) = \lambda_0 \cdot f^b = f^b 10^{-6}$. Figure 2a shows (1−*performability*) for different frequencies with $b = 0.1, 1$ and 10, respectively. The results coincide with our analysis in Section IV, that is, the performability of an application decreases monotonically when $b \leq 1$ and may increase if lower frequencies result in the same number of recoveries when $b > 1$. Each sharp jump in the figure reflects one fewer recovery being scheduled due to lower frequencies. In general, when b is small, the number of recoveries determines the level of performability with small variations. However, for larger b (e.g., $b = 10$), lower frequencies may result in higher performability for the same number of recovery sections due to fast decreased fault rate.



a. 1−*performability* b. energy consumption

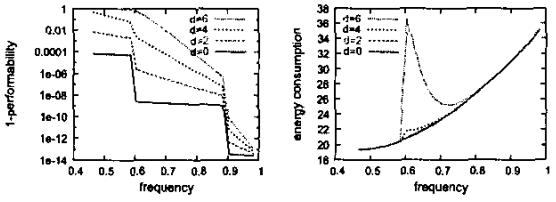
Fig. 2. Effects of frequency scaling: a. the performability for different values of b; b. the expected energy consumption for different frequency-independent active power P_{ind}.

To see the effects of frequency-independent active power on the energy consumption under frequency scaling, with fixed $b = 1$ (i.e., strict linear relation between the fault rate and frequency), Figure 2b shows the expected energy consumption for the application running at different frequencies when $P_{ind} = 0$, 0.2 and 0.4, respectively. When there is no frequency-independent active power (i.e., $P_{ind} = 0$), the expected energy consumption is constant since the energy consumption for original execution is constant (recall that voltage is fixed as $V_{max} = 1$) and the probability of executing the recoveries is rather small (less than 0.01%, see Figure 2a). However, for the cases of $P_{ind} = 0.2$ and $P_{ind} = 0.4$, the expected energy consumption increases when frequency decreases due to increased frequency-independent energy consumed. The results also confirm the analysis in Section IV.

Therefore, we conclude that, when it is possible to efficiently remove the frequency-independent active power by putting systems to sleep, scaling down the frequency with fixed supply voltage increases energy consumption as well as decreases the performability and should not be employed.

B. Voltage Scaling with Exponentially Increased Fault Rates

To show the significance of fault rate changes on performability under voltage scaling, we vary the value of d and the results are shown in Figure 3. Recall that the fault rate under frequency f and voltage V $(= V_{max}\frac{f}{f_{max}} = f)$ is $\lambda(f) = \lambda_0 \cdot 10^{\frac{d(1-f)}{1-f_{min}}}$ and $\lambda_0 = 10^{-6}$. For $P_{ind} = 0.2$, we have $f_{min} \approx 0.46$ (see Section II). From Figure 3a, we can see that larger values of d (i.e., faster fault rate increases) lead to worse performability for lower frequencies. The sharp step decreases in the performability correspond to one fewer recovery section being scheduled because of reduced frequency. The performability for fixed fault rate ($d = 0$) is much better than that of variable fault rates (e.g., better than the case of $d = 2$ in almost 2 orders of magnitude). Therefore, in general, the number of recovery sections determines the level of performability and assuming fixed fault rate ($d = 0$) when reducing voltages and frequencies may result in unsatisfied performability.



a. 1−*performability* b. energy consumption

Fig. 3. The performability and expected energy consumption for different values of d.

Figure 3b shows the corresponding expected energy consumption for different values of d. In general, when frequency and voltage decrease, the expected energy consumption decreases, which illustrates the trade-off between performability and expected energy consumption. However, the expected energy consumption for the case of $d = 6$ increases at lower frequencies when $0.60 \leq f \leq 0.71$. The abnormality comes from the high probability (close to 100% when $f = 0.60$) of recovery sections being executed when $d = 6$, which overshadows the energy saved during original execution. The sharp decreases around $f = 0.59$ comes from no recovery being scheduled within the application deadline.

The results show the implications of the applicability of voltage scaling when performability is a major concern. For example, if the required performability goal is $1 - 10^{-7}$, the frequency and the corresponding voltage can be reduced to 0.6 when ignoring the effect of voltage scaling on fault rates (i.e., assuming $d = 0$). However, when considering fault rate changes caused by voltage scaling, we can only reduce the frequency and the corresponding voltage to 0.78 when $d = 2$ or 0.89 when $d = 4$. Thus, the previous work that ignores the effect of voltage scaling on fault rates is too optimistic and may lead to unsatisfied performability when reducing frequency and voltage for energy savings [27].

VI. Conclusions

This work explores the effects of energy management on system reliability for real-time embedded systems. In addition to being used by frequency and voltage scaling to save energy, the slack time in real-time systems can also be used by roll-back recovery to increase system reliability. Considering the rate of transient faults (i.e., soft errors caused, for example, by cosmic ray radiations) also depends on system operating frequency and supply voltage, the trade-off between system reliability and energy consumption becomes more complicated. As the first attempt to incorporate the effects of frequency and voltage scaling on fault rates, we consider two fault rate models based on previously published results, specifically, the linearly decreasing fault rate model under frequency scaling (with fixed

supply voltage) and the exponentially increasing fault rate model under voltage scaling (i.e., reducing supply voltage for lower frequencies).

Our analysis results show that, in general, system reliability depends largely on the number of recoveries allowed within an application's deadline. Under frequency scaling (with fixed supply voltage), more energy is consumed due to static power and lower reliability is obtained due to fewer recoveries at lower frequencies, especially when the fault rate decreases linearly or sub-linearly with reduced frequency. When the fault rate decreases super-linearly, lower frequencies result in higher reliability if the same number of recoveries is obtained. For voltage scaling, energy is saved at the expense of significantly reduced reliability, especially when the exponential increasing fault rate model is considered. The expected energy consumption may increase if the high fault rate at lower frequency and voltage leads to dramatically high probability of executing the recovery sections, which overshadows the energy savings obtained via voltage scaling.

The above conclusions are based on the specific fault rate models assumed and different fault rate models may lead to different conclusions. However, the main contribution of this work is to show that energy management through frequency and voltage scaling has significant effects on system reliability. Thus, for critical applications, the goal of saving energy by reducing system supply voltage and processing frequency must be carefully weighted with the goal of maintaining a certain level of reliability. Ignoring the effects of energy management on fault rate is too optimistic and may lead to unsatisfied reliability goals.

ACKNOWLEDGMENTS

This work has been supported by the Defense Advanced Research Projects Agency through the PARTS (Power-Aware Real-Time Systems) project (Contract F33615-00-C-1736).

REFERENCES

[1] S. Buchner, M. Baze, D. Brown, D. McMorrow, and J. Melinger. Comparison of error rates in combinational and sequential logic. *IEEE Trans. on Nuclear Science*, 44(6):2209–2216, 1997.

[2] T. D. Burd and R. W. Brodersen. Energy efficient cmos microprocessor design. In *Proc. of The HICSS Conference*, Jan. 1995.

[3] X. Castillo, S. McConnel, and D. Siewiorek. Derivation and caliberation of a transient error reliability model. *IEEE Trans. on computers*, 31(7):658–671, 1982.

[4] Intel Corp. Mobile pentium iii processor-m datasheet. Order Number: 298340-002, Oct 2001.

[5] E. (Mootaz) Elnozahy, M. Kistler, and R. Rajamony. Energy-efficient server clusters. In *Proc. of Power Aware Computing Systems*, 2002.

[6] E. (Mootaz) Elnozahy, R. Melhem, and D. Mossé. Energy-efficient duplex and tmr real-time systems. In *Proc. of The* 23^{rd} *IEEE Real-Time Systems Symposium*, Dec. 2002.

[7] X. Fan, C. Ellis, and A. Lebeck. The synergy between power-aware memory systems and processor voltage. In *Proc. of the Workshop on Power-Aware Computing Systems*, 2003.

[8] K. J. Hass, J. W. Gambles, B. Walker, and M. Zampaglione. Mitigating single event upsets from combinational logic. In *Proc. of the* 7^{th} *NASA Symposium on VLSI Design*, 1998.

[9] P. Hazucha and C. Svensson. Impact of cmos technology scaling on the atmospheric neutron soft error rate. *IEEE Trans. on Nuclear Science*, 47(6):2586–2594, 2000.

[10] T. Ishihara and H. Yauura. Voltage scheduling problem for dynamically variable voltage processors. In *Proc. of The 1998 International Symposium on Low Power Electronics and Design*, Aug. 1998.

[11] T. Juhnke and H. Klar. Calculation of the soft error rate of submicron cmos logic circuits. *IEEE Journal of Solid-State Circuits*, 30(7):830–834, 1995.

[12] K. M. Kavi, H. Y. Youn, and B. Shirazi. A performability model for soft real-time systems. In *Proc. of the Hawaii International Conference on System Sciences (HICSS)*, Jan. 1994.

[13] H. Lee, H. Shin, and S. Min. Worst case timing requirement of real-time tasks with time redundancy. In *Proc. of Real-Time Computing Systems and Applications*, 1999.

[14] P. Liden, P. Dahlgren, R. Johansson, and J. Karlsson. On latching probability of particle induced transients in combinational networks. In *Proc. of the* 24^{th} *International Symposium on Fault-Tolerant Computing*, 1994.

[15] R. Melhem, D. Mossé, and E. (Mootaz) Elnozahy. The interplay of power management and fault recovery in real-time systems. *IEEE Trans. on Computers*, 53(2):217–231, 2004.

[16] A. Miyoshi, C. Lefurgy, E. V. Hensbergen, R. Rajamony, and R. Rajkumar. Critical power slope: Understanding the runtime effects of frequency scaling. In *Proc. of the ACM International Conference on Supercomputing*, 2002.

[17] T. Pering, T. Burd, and R. Brodersen. The simulation and evaluation of dynamic voltage scaling algorithms. In *Proc. of Int'l Symposium on Low Power Electronics and Design*, Aug. 1998.

[18] D. K. Pradhan. *Fault Tolerance Computing: Theory and Techniques*. Prentice Hall, 1986.

[19] Rambus. Rdram. http://www.rambus.com/, 1999.

[20] N. Seifert, D. Moyer, N. Leland, and R. Hokinson. Historical trend in alpha-particle induced soft error rates of the alpha^{TM} microprocessor. In *Proc. of the* 39^{th} *Annual International Reliability Physics Symposium*, 2001.

[21] Tezzaron Semiconductor. Soft errors in electronic memory: A white paper. available at http://www.tachyonsemi.com/about/papers/, 2004.

[22] K. Seth, A. Anantaraman, F. Mueller, and E. Rotenberg. Fast: Frequency-aware static timing analysis. In *Proc. of the* 24^{th} *IEEE Real-Time System Symposium*, 2003.

[23] P. Shivakumar, M. Kistler, S. W. Keckler, D. Burger, and L. Alvisi. Modeling the effect of technology trends on the soft error rate of combinational logic. In *Proc. of the International Conference on Dependable Systems and Networks*, 2002.

[24] A. Sinha and A. P. Chandrakasan. Jouletrack - a web based tool for software energy profiling. In *Proc. of Design Automation Conference*, Jun 2001.

[25] O. S. Unsal, I. Koren, and C. M. Krishna. Towards energy-aware software-based fault tolerance in real-time systems. In *Proc. of The International Symposium on Low Power Electronics Design (ISLPED)*, Aug. 2002.

[26] M. Weiser, B. Welch, A. Demers, and S. Shenker. Scheduling for reduced cpu energy. In *Proc. of The First USENIX Symposium on Operating Systems Design and Implementation*, Nov. 1994.

[27] D. Zhu, R. Melhem, D. Mossé, and E.(Mootaz) Elnozahy. Analysis of an energy efficient optimistic tmr scheme. In *Proc. of the* 10^{th} *International Conference on Parallel and Distributed Systems (ICPADS)*, Jul. 2004.

[28] J. F. Ziegler. Terrestrial cosmic ray intensities. *IBM Journal of Research and Development*, 42(1):117–139, 1998.

[29] J. F. Ziegler. Trends in electronic reliability: Effects of terrestrial cosmic rays. available at http://www.srim.org/SER/SERTrends.htm, 2004.