Lifetime Optimization for Real-Time Embedded Systems Considering Electromigration Effects

Taeyoung Kim*, Bowen Zheng†, Hai-Bao Chen†, Qi Zhu†, Valeriy Sukharev‡ and Sheldon X.-D. Tan†
* Department of Computer Science and Engineering, University of California, Riverside, CA 92521
† Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521
‡ Mentor Graphics Corporation, Fremont, CA 94538, USA

***Abstract*—In this article, we propose a new lifetime task optimization techniques for real-time embedded processors considering the electromigration-induced reliability. The new approach is based on a recently proposed physics-based electromigration (EM) model for more accurate EM assessment of a power grid network at the chip level. We apply the dynamic voltage and frequency scaling (DVFS) (by selecting the performance states or p-states of the tasks to manage the power and thus the lifetime of the processor running different tasks over their periods. We consider both single-rate and multi-rate embedded systems with preemption. To model the mean-time-to-failure (MTTF) of a task for a given p-state, response surface modeling is applied. We then frame the reliability optimization problem as the continuous constrained nonlinear optimization problem in which the system EM-induced reliability is maximized subject to the timing constraints, which is further solved by simulated annealing method. Experimental results show that for low utilization systems, significant reliability improvement can be achieved with even smaller power consumption than existing reliability-ignore scheduling method. The proposed can lead to near Pareto's front trade-off between the power/energy and the lifetime compared to the existing task scheduling method.**

I. Introduction

Long-term reliability is becoming a limiting constraint for high performance and embedded real-time system designs due to the high failure rates in deep submicron and nanoscale devices. The increase in failure rates is caused by high integration levels and higher power densities, which leads to excessive on-chip temperatures. The introduction of new materials, processes and devices, coupled with voltage scaling limitations and increasing power consumption will impose many new reliability challenges. The semiconductor industry faces the challenges to maintaining reliability such as the continued increase in die size and number of transistors and the constant scaling of transistors for performance [1]. Increasing transistor density and thus power density is causing higher temperatures on chip, resulting in failure acceleration. Scaling to smaller transistors increases failure rates by shrinking the thickness of dielectrics. This has led the International Technology Roadmap for Semiconductor (ITRS) to predict the onset of significant reliability problems in the future, and at a pace that has not been seen in the past [11]. Furthermore, for safety-critical real-time embedded systems (such as satellite and surveillance systems) where reliability is as important as energy efficiency, reliability-aware energy management becomes a necessity.

This work is supported in part by NSF grant under No. CCF-1255899, in part by Semiconductor Research Corporation(SRC) Grant under No. 2013-TJ-2417 and in part by Academic Senate COR Fellowship.

Some initial efforts have been carried out for system level reliability analysis for SoCs (system-on-a-chip). RAMP [19] is the first architecture level tool for modeling the long-term processor reliability of microprocessors at the design stage. The follow-up work by the same authors proposed a dynamic reliability management (DRM) concept by dynamic voltage and frequency scaling (DVFS) [18]. It showed that it was not sufficient to just manage the temperature or power from the reliability perspective. For real-time embedded systems, many existing works focus on minimizing energy consumption while meeting all the deadlines for various real-time task models. Existing works include power management schemes, which exploits the available static and/or dynamic slack in the systems [2], [5], [17]. For many-long term reliability effects, reducing power will implicitly improve the reliability of a processor. But the two objectives are still not the same. Some reliability-aware power management works have been proposed recently [16], [22] by using low power techniques such as DVFS. But most of those existing works focus on the transient errors instead of long-term wearout failures. Recently a reliability-aware task allocation and scheduling method multi-core embedded processors were proposed [7]. This work considers long-term failure mechanisms using a general reliability models. However, such general models will not be accurate for specific failure mechanisms. Also task allocation and scheduling is not nest knobs to manage the long-term wearout failures as they will not significantly change the temperatures of the chip as our study shows. Low power techniques like DVFS is more desired.

In this article, we propose a new lifetime task optimization techniques for real-time embedded processors considering the electromigration-induced reliability. The new approach is based on a recently proposed physics-based electromigration (EM) model for more accurate EM assessment of a power grid network at the chip level. We apply the dynamic voltage and frequency scaling (DVFS) (by selecting the performance states or p-states of the tasks to manage the power and thus the lifetime of the processor running different tasks over their periods. We consider both single-rate and multi-rate embedded systems with preemption. To model the mean-time-to-failure (MTTF) of a task for a given p-state, response surface modeling is applied. We then frame the reliability optimization problem as the continuous constrained nonlinear optimization problem in which the system EM-induced reliability is maximized subject to the timing constraints, which is further solved by simulated annealing method. Experimental results show that for low utilization systems, significant reliability improvement can be achieved with even smaller power consumption than existing reliability-ignore scheduling method. The proposed can lead to near Pareto's front trade-off between the power/energy and the lifetime compared to the existing task scheduling method.

II. New physics-based EM modeling and analysis

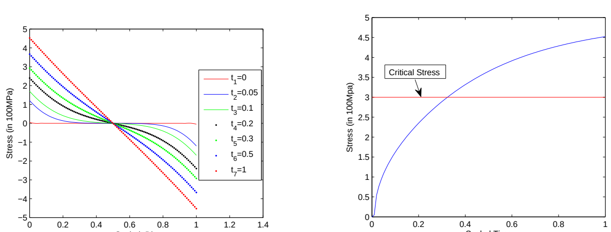
EM is a physical phenomenon of the migration of metal atoms along a direction of applied electrical field. Atoms (either lattice atoms or defects/impurities) migrate toward the anode end of metal wire along the trajectory of conducting electrons. Over time, the lasting unidirectional electrical load increases these stresses, as well as the stress gradient along the metal line. In some cases, usually when a line is long, this stress can reach a critical level, resulting in a void nucleation at the cathode and/or hillock formation at the anode end of line.

A. Void dynamics: Nucleation and growth phases

Development of such analytical formulation was proposed by Korhonen [12] Since the atomic flux divergence results in the volumetric strain, the derived one dimensional diffusion-like equation for the hydrostatic stress field $\sigma(x,t)$, takes the form, [12]:

$$\frac{\partial \sigma}{\partial t} = \frac{\partial}{\partial x}\left[\kappa\left(\frac{\partial \sigma}{\partial x} + \frac{eZ\rho j}{\Omega}\right)\right] \quad (1)$$

Here, $\kappa = D_a B\Omega/kT$, where D_a is the atomic diffusivity, and B is the bulk modulus, Ω is the atomic volume; e is the electron charge, eZ is the effective charge of the migrating atoms, ρ is the wire electrical resistivity. Solution of this initial-boundary value problem is the infinite series [12].



(a) (b)

Fig. 1. (a) EM-stress distribution change over time in a simple metal wire. (b) EM-stress evaluation versus time.

Fig. 1(a) shows the EM-induced stress development for a single metal wire over time from the finite element analysis for a given current density and temperature setting. Fig. 1(b) shows the stress evaluation over time. Each time unit here is 10^7 seconds. During this process, tensile stress (positive stress) will be developed at the cathode side (the left node), while compressive stress (negative stress) will be generated at the anode side of the wire (right node). When the tensile stress hits critical stress, void will be generated, which is called nucleation time. After this, the void will grow, which will increase the resistance of the wires until the growth stops. In contrast, existing EM model only consider one of such two processes modeled by Black's equation. Also when the wire hits its MTTF, the wire is treated as open circuit, which will over-estimate EM-impacts in circuit reliability.

Since finite element method can't scale very well, a more physics-based compact EM model has been proposed recently for full-chip reliability analysis [8], [21], which is the basis for the proposed work. In this new EM model, the EM development process consists of two phases - the nucleation phase and the growth phase. In the first nucleation phase, a closed-form expression to compute the nucleation time (t_{nuc}) is given, which is a function of current density, temperature, the residual stress of the wire due to thermal and other effects as well as other wire geometry and material parameters. Approximate value of void nucleation time (t_{nuc}) extracted from this solution, which is determined as an instant in time when stress at the cathode end of the line reaches σ_{crit}, corresponds well to an analytical formulation of t_{nuc} derived from the approximate solution of continuity equations for evolution of vacancy and plated atom concentrations (see, for example [20]) in the confined 1D line

$$\begin{aligned} t_{nuc} \approx & \tau^{\star} e^{\frac{E_V}{kT}} e^{-\frac{f\Omega}{kT}\left(\sigma_{Res}+\frac{eZ\rho l}{4\Omega}j\right)} \\ & ln\left\{\frac{\frac{eZ\rho l}{4\Omega}j}{\sigma_{Res}+\frac{eZ\rho l}{4\Omega}j-\sigma_{CR}}\right\} \end{aligned} \quad (2)$$

where $\tau^{\star} = \frac{l^2}{D_0} e^{E_D/kT} \frac{kT}{\Omega B}$. Here, j is the current density, T is temperatures, k_B is the Boltzmann's constant, l is the segment length, E_V and E_D are the activation energy of vacancy formation and diffusion, f is the ratio of volumes occupied by vacancy and lattice atom, σ_{crit} is the critical stress needed for the failure precursor nucleation (void/hillock). σ_{Res} is the residual stress of the metal segment from the cooling process and other factors.

The second phase is the void size growth: voids are formed at t_{nuc} and grow at $t > t_{nuc}$. The wire resistance starts to increase over the time in the growth phase. As a result, the p/g network becomes a time-varying network and its voltage drops will keep changing over the time [8].

B. EM assessment at power grid level

Because of the concern with the long-term average effects of the current, in EM related work a DC model of the power grid is generally assumed [4]. As a result, we consider only the EM-induced kinetics of the power grid network resistances. In our problem formulation, each mortal wire, which subjects to the EM impact, will start to change its resistance value upon achieving the nucleation time. As a result, we end up with the power grid systems, which is a linear, time-varying and driven by the DC effective currents, which is modeled as $G(t)v(t) = I_{eff}$, where, $G(t)$ a $n \times n$ time-varying conductance matrix; I_{eff} is the effective DC current source vector; $v(t)$ is the corresponding vector of nodal voltages and n is the size unknown voltages. In our problem, the time scale is the EM time scale, which can be months or years.

In the new EM-induced reliability analysis algorithm for p/g networks, we compute the voltage drops of the grids at fixed EM time step. The resistance of one or more wires begins to change (increase) starting with their nucleation times. At each time step, we collect new wires whose nucleation times were reached, and compute the new resistance for existing wires in the growth phases and corresponding voltage drops of the whole grids. This process is repeated until the voltage drop of one or more nodes exceeds the critical voltage drops allowed (say 10% of Vdd).

C. System level EM-reliability model

At the system level, the embedded system will run on different tasks under different p-states. As a result, its temperature and current densities will change with time. However existing EM models including the new physics-based model can only take constant temperature. Previous study shows that whole system MTTF or lifetime under different temperature can be approximated by [15]:

$$lifetime = \frac{1}{(\sum_{k=1}^{n}(\Delta t_k \frac{1}{MTTF_{R,k}}))/T} \tag{3}$$

where $MTTF_{R,k}$ is the actual MTTF under the k-th power and temperature settings for Δt_k period, assuming the chip works through n different power and temperature settings and $T = \sum_{k=1}^{n} \Delta t_k$. Each $MTTF_{R,k}$ will be computed based on the EM models discussed in the previous section.

III. New lifetime optimization method

In this section, we introduce lifetime optimization method for real-time embedded systems considering EM effects. To effectively manage the lifetime of the chips due to EM effect, we still apply the dynamic voltage and frequency scaling (DVFS), which is implemented by performance-state (p-state) selections for a processor [6]. We assume that a task scheduling for an embedded system has been finished before the optimization. In this section, we first present the real-time task models considered in this work.

A. Real-time system models

Most of embedded systems are real-time systems, where tasks are activated periodically, so timing and deadline should be carefully considered. We mainly consider two kinds of models for real-time systems: (1) Single-rate model, where all tasks in the system have the same activation period and deadline, and (2) Multi-rate model, where each task can have its own activation period and deadline. A task set is represented as $\mathcal{T} = \{\tau_1, \tau_2, \tau_3, \dots, \tau_n\}$ where all tasks in the system are independent of each other. All tasks are scheduled on a single-core processor. For single-rate system, every task has the same activation period T. For multi-rate system, every task has its period and we use T_i to denote the period of task τ_i. In this work, we consider the deadline of each task as its activation period, where every task must finish the execution before its next activation, or missing deadline may have detrimental impacts on the whole system.

In *single rate system*, every task has the same period and deadline, so the worst case for timing is that one task has to wait for all the other tasks to finish execution. As long as the sum of all task execution times are no greater than deadline T, the system is schedulable. We use Δt_i to represent the execution time of each task, so the timing constraints for single-rate system is expressed as $\sum_i \Delta t_i \leq T$.

In *multi rate system*, the task scheduling is repeated at a hyper period that is the LCM (least common multiple) of all periods, so we need to consider the p-state selection to tasks within a hyper period for multi-rate system. Thus, we assume that for every activation the task uses the same p-state. We apply a fixed-priority scheduling method, in which higher priority tasks can preempt lower priority tasks, so the order of tasks is obtained from the priority. This scheduling method is widely used and is supported by standards like OSEK [3].

Task response time is an important metric to analyze timing in multi rate system. It represents the time duration from the activation of one task to its termination. In fixed-priority scheduling, higher priority tasks can preempt lower priority tasks, so the *response time* of one task contains the time it is preempted by higher priority tasks besides its execution time. We use r_i to represent the response time of task τ_i and it is expressed as (4) where $hp(\tau_i)$ denotes the task set containing higher priority tasks than task τ_i. The first term of (4) represents the execution time of task τ_i and the second term denotes the preemption time of higher priority tasks.

$$r_i = \Delta t_i + \sum_{\tau_k \in hp(\tau_i)} \lceil \frac{r_i}{T_k} \rceil \Delta t_k \tag{4}$$

The timing constraints for multi-rate system is that the response time of any task should be no greater than its period: $\forall i : r_i \leq T_i$. Every task must finish its execution before its next activation. In our real-time system model, we use *Rate Monotonic scheduling*, which is a common preemptive fixed-priority scheduling method. In [14], it is proved that Rate Monotonic is optimal with respect to feasibility among fixed priority single-processor schedulers. As an illustration, a real-time system model for single and multi rate is shown in Fig. 2 where task 2 has the highest priority due to its smallest period and task 3 has been preempted two times by task 2 within its response time. Further, we use *core utilization* to measure

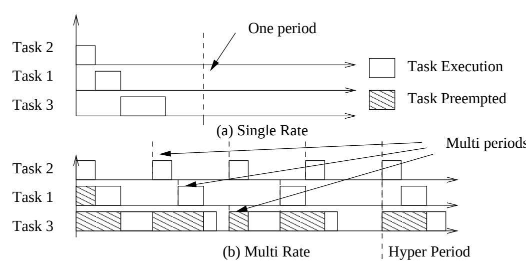


Fig. 2. Single rate and multi rate task scheduling models

the percentage of total execution time in the processor, which has significant impacts on the lifetime improvement we can obtain. The core utilization u is expressed as $u = \sum_i \frac{\Delta t_i}{T_i}$, where Δt_i is the execution time for task i and T_i is the period of task.

B. The new lifetime optimization algorithm for real-time embeded systems

In this section, we explain the new algorithm and its major steps. The algorithm flow is shown in Algorithm 1. First, we start with a single or multi rate task set in either single-rate or multi-rate models. The tasks have non-optimized initial p-state, which has a pair of highest operating frequency and voltage. The temperature values from the profiled power and execution time can be measured by running HotSpot [13] with respect to every possible p-state selection for each task.

Once we have all the power and temperature information, we compute the MTTF for each p-state for the task based on the after-mentioned EM simulator discussed in the previous section. As a result, for each p-state p_j of a task i, the execution time Δ_{t_i} is the the function of its $MTTF_i = MTTF(p_j)$ under the p-state p_j. We then build a continuous function of $\Delta_{t_i}(MTTF_i)$ using the response surface method (RSM). We may use up to 3rd order polynomials for our RSM method. The function is important for the lifetime optimization as shown later. Then we solve a constrained nonlinear optimization problem to find the best p-state for each task.

Algorithm 1 EM-induced lifetime optimization algorithm

Input: A task set with execution time, power, and available p-states

Output: Optimized p-state selection for a task set

1: Compute scaling voltage and frequency for each task with every p-state
2: Compute a period or hyper period for single or multi rate system
3: Compute scaled power and measured temperature for each task with every p-state
4: With given power and temperature, $MTTF_i = MTTF(p_j)$ can be calculated for each p-state p_j.
5: Calculate the task execution time and MTTF for each task for every p-state. We then build continous function of $\Delta_{t_i}(MTTF_i)$ for each task i.
6: Perform the MTTF optimization as shown in (5) with timing constraint to find best p-state for each task.
7: Output each selected p-state for each task.

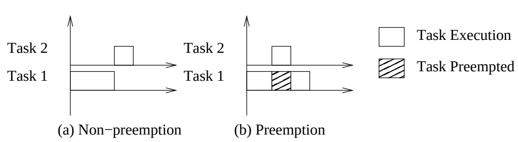


Fig. 3. Multi rate preemption

1) Multi rate preemption impacts on EM reliability: For a multi rate system, one important problem is that preemption can lead to interruption of a task execution. Fig. 3 shows an example of the mechanism of preemption in which task 1 has been preempted with task 2. Since each task has its power (thus temperature profile), does such re-ordered task execution will affects the EM-induced reliability? It turns out that such task re-ordering or task preemption has marginal impacts on the EM-induced reliability. Table I show the results for the MTTF of two executions: one for non-preemption and one for preemption. As we can see that the MTTFs for both cases are almost same. The reason is that the temperature of the each task execution is mainly determined by its power, its transient thermal effects between task transition are not significant. Thus, the lifetime formulation in (3) remains the same for multi-rate system.

2) MTTF optimization formulation: Now, we can formulate the MTTF optimization based on all the previous discussions.

TABLE I

A PRELIMINARY MEASUREMENT FOR PREEMPTIVE EFFECT ON LIFETIME

Non-preemption	Task1	Task2	Lifetime
$MTTF_i$	0.47	83	0.71
Preemption	Task1 (Divided)	Task2	Lifetime
$MTTF_i$	0.47,0.47	84.3	0.71

The whole optimization can be formulated in (5).

$$\text{Max } Lifetime(m) = \frac{1}{(\sum_{n}^{i=1}(\Delta t_i \frac{1}{MTTF_i} \frac{T_{hyper}}{T_i}))/T_{hyper}}$$

Subject to:

$$r_i = \Delta t_i + \sum_{k \in hp(i)} \left\lceil \frac{r_i}{T_k} \right\rceil \Delta t_k \leq T_i$$
$$MTTF_{i,l} \leq MTTF_i \leq MTTF_{i,u} \quad (5)$$

where $m = [MTTF_1, MTTF_2, \ldots, MTTF_n]^T$, which is the variable vector, i is the task id, Δt_i is the execution time for task i , n is the total number of task, $MTTF_i$ is the segment MTTF for task i r_i is the response time of task i, $hp(i)$ is the task set containing the higher priority tasks than the current task i. T_i is the period of task i, which is a deadline. T_{hyper} is the hyper period for all tasks. At the single rate T_{hyper} is equal to T_i. T_{total} is the total execution time of all tasks. $MTTF_{i,l}$ is the minimum bound of MTTF for task i and $MTTF_{i,u}$ is the maximum bound of MTTF for task i.

3) Optimization solution: To solve the constrained nonlinear optimization problem, the simulated annealing method is applied. We use the Matlab's global optimization toolbox, which provides the simulated annealing function code based on adaptive simulated annealing (ASA) [9]. The simulated annealing begins with an initial $MTTF_i$ obtained by a median of $MTTF_{i,l}$ and $MTTF_{i,u}$ for each task i. The algorithm allows a large number of moves to gradually improve MTTF. The step length equals the current temperature, and the moving direction is uniformly random. The problem type is set as *bound constrained* with a set of bounds, which are $MTTF_{i,l}$ and $MTTF_{i,u}$ for each task i. The regression coefficients in our RSM model can be parameterized by the variables ($MTTF_i$) of our objective function and act as constants during the optimization. The temperature will be lowered by 0.95^k at each iteration, where k is annealing parameter, which is the same as the iteration number before the reannealing. The stopping criterion is set to 10^{-6}, where the iteration stops when the average lifetime variation in the objective function is smaller than this tolerance. The maximum number of evaluation is set to 3000. Once a solution m is found, the corresponding p-state p_j for each $MTTF_i$ for each task is calculated by finding the shortest-distance between $MTTF_i(p_j)$ and computed $MTTF_i$.

IV. EXPERIMENTAL RESULT AND DISCUSSIONS

A. Experimental setup

The proposed new lifetime task optimization method has been implemented in Matlab and C++. We applied the recently proposed physics-based EM model and analysis method for

TABLE II

OPTIMIZATION METHOD EVALUATION FOR LOW/HIGH CORE UTILIZATION SINGLE RATE TASK

Non optimized	6 Tasks at 40% core utilization at single rate					
i (task id)	1	2	3	4	5	6
P_k (p-state)	1	1	1	1	1	1
Δt_i(ms)	3.70	5.82	2.43	5.10	0.92	1.99
$Period_i$ (ms)	50	50	50	50	50	50
MTTF (year)	0.47	0.47	0.46	0.47	0.47	0.47
Energy (Wh)	0.11	0.17	0.07	0.15	0.02	0.05
Total Energy	**0.59**		**Lifetime**		**1.19**	
Optimized						
i (task id)	1	2	3	4	5	6
P_k (p-state)	5	5	5	5	5	5
Δt_i (ms)	6.32	9.93	4.15	8.69	1.57	3.40
$Period_i$ (ms)	50	50	50	50	50	50
$MTTF_i$ (year)	44.8	44.7	48	44.7	50	48
$Energy_i$ (Wh)	0.07	0.11	0.04	0.10	0.01	0.04
Total Energy	**0.40**		**Lifetime**		**66.97**	

Non optimized	6 Tasks at 80% core utilization at single rate					
i (task id)	1	2	3	4	5	6
P_k (p-state)	1	1	1	1	1	1
Δt_i(ms)	6.81	5.49	8.37	5.25	7.45	6.51
$Period_i$ (ms)	50	50	50	50	50	50
$MTTF_i$ (year)	0.47	0.47	0.46	0.47	0.47	0.47
$Energy_i$ (Wh)	0.20	0.16	0.25	0.15	0.22	0.19
Total Energy	**1.19**		**Lifetime**		**0.59**	
Optimized						
i (task id)	1	2	3	4	5	6
P_k (p-state)	2	3	3	3	2	3
Δt_i (ms)	7.79	7.18	10.93	6.86	8.52	8.50
$Period_i$ (ms)	50	50	50	50	50	50
$MTTF_i$ (year)	0.95	3.06	3.06	3.06	0.95	3.06
$Energy_i$ (Wh)	0.18	0.13	0.20	0.13	0.20	0.16
Total Energy	**1.02**		**Lifetime**		**1.78**	

TABLE III

OPTIMIZATION METHOD EVALUATION FOR LOW/HIGH CORE UTILIZATION MULTI RATE TASK

Non optimized	6 Tasks at 40% core utilization at multi rate					
task id i	1	2	3	4	5	6
P_k (p-state)	1	1	1	1	1	1
Δt_i (ms)	6.93	4.02	4.19	1.91	1.64	3.03
$Period_i$ (ms)	100	100	50	20	20	100
$MTTF_i$ (year)	0.47	0.47	0.46	0.47	0.47	0.47
$Energy_i$ (Wh)	0.20	0.12	0.25	0.28	0.24	0.09
Total Energy	**1.20**		**Lifetime**		**1.18**	
Optimized	6 Tasks at 40% core utilization at multi rate					
task id i	1	2	3	4	5	6
P_k (p-state)	5	5	5	5	5	5
Δt_i (ms)	11.8	6.58	7.15	3.27	2.79	5.18
$MTTF_i$ (year)	44.7	44.8	44.8	48.2	49	48
$Energy_i$ (Wh)	0.14	0.08	0.17	0.19	0.16	0.06
Total Energy	**0.82**		**Lifetime**		**68.02**	

Non optimized	6 Tasks at 80% core utilization at multi rate					
i (task id)	1	2	3	4	5	6
P_k (p-state)	1	1	1	1	1	1
Δt_i (ms)	9.33	7.25	16.38	13.94	10.05	3.30
$Period_i$ (ms)	100	50	100	100	100	20
MTTF (year)	0.47	0.47	0.46	0.47	0.47	0.47
Energy (Wh)	0.27	0.43	0.49	0.41	0.30	0.49
Total Energy	**2.42**		**Lifetime**		**0.58**	
Optimized	6 Tasks at 80% core utilization at multi rat					
i (task id)	1	2	3	4	5	6
P_k (p-state)	3	4	3	1	2	2
Δt_i (ms)	12.18	10.82	21.39	13.94	11.49	3.77
MTTF (year)	3.06	15.2	3.06	0.47	0.95	0.95
Energy (Wh)	0.23	0.32	0.40	0.41	0.27	0.45
Total Energy	**2.10**		**Lifetime**		**1.36**	

our EM analysis [8]. Hotspot [13] is used for the temperature modeling.To generate task sets, we implement a random real-time task generator based on the core utilization factor. We use 6 tasks per one task set and 36 task set. For single rate task sets, 50ms, 20ms, and 10ms are chosen for a single rate period. For multi rate task sets, we use 50ms, 20ms, and 10ms periods, but set to the same hyper period as 100ms. All benchmarks and Matlab environment are running on 4 core 3.0Ghz Xeon server with 16GB RAM running Linux. 5 p-states $p_j = \{$(1.6Ghz, 1.484V),(1.4Ghz, 1.409V), (1.22Ghz, 1.339V), (1.07Ghz, 1.272V), (930Mhz, 1.208V)$\}$ are chosen from ACPI standard and Enhanced Intel Speedstep Technology [6], [10].

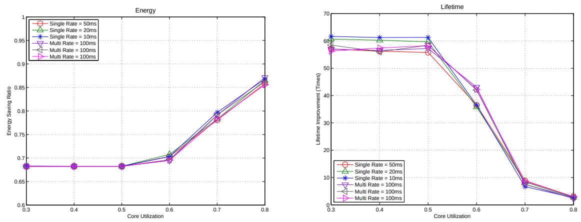
B. *Evaluation of proposed lifetime optimization*

First, we evaluate our lifetime optimization method (see Section III) by comparing its lifetime and energy with non-optimized p-state tasks to the optimized p-state tasks. In the evaluation of our lifetime optimization, we consider four different task sets (low and high core utilization, and single and multi rates) for comparison. Table II summarize the results for 40% and 80% core utilization of single rate real-time tasks. As we can see, for the low core utilization tasks, the optimization method finds the lowest-energy p-state solution for each task and total energy consumption decreases about 67% as each task p-state is selected as 5, which is the lowest-engery DVFS for each task execution. With this optimization method, the lifetime is improved to 66.97 year from 1.19 year. But for high utilization case, lifetime improvement will be limited as shown in the 80% utilization case. In this case, the most of the p-states are moved to middle range (3 in this case). But still there is still 85% energy saving and 3X lifetime improvement achieved by the proposed lifetime optimization method. This indicates the significant improvement can be made for both energy and reliability from the simple task scheduling.

In addition to single rate task, we also show results in multi rate tasks summarized in Table III. Again, depending on the core utilization, the energy and lifetime can be both improved and the improvement can be noticeable or significant.

C. *Core utilization effects and trade-off on energy and lifetime*

As we can see, core utilization factor can have significant impacts on the final results. Figure 4 shows the core utilization versus energy and lifetime. The experiments are simulated with single rate tasks at 50ms, 20ms, and 10ms periods, and 3 random multi rate tasks at 100ms hyper period. In the low core utilization, the system can run the task under higher p-state as the core utilization is the ratio of the task execution time and its period. In the higher utilization, energy saving and lifetime improvements decrease as the higher utilization leads less number of p-state selection.



(a) (b)

Fig. 4. Core utilization effects on (a) energy saving and (b) lifetime improvement

We also observe the Pareto-like trade-off between the energy and lifetime obtained from the proposed lifetime optimization as shown in Figure 5. This curve is obtained for different core utilizations, which determine the potential for lifetime improvement and energy reduction. So in general, low energy will lead to better lifetime and the proposed method can lead to the best (Pareto-like) trade-off.

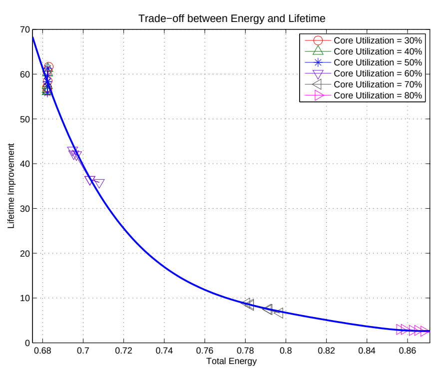


Fig. 5. Trade-off between lifetime and energy

V. Conclusion

We have proposed a new lifetime task optimization techniques for real-time embedded processors considering the electromigration-induced reliability. The new approach was based on a recently proposed physics-based electromigration (EM) model for more accurate EM assessment of a power grid network at the chip level. We applied the dynamic voltage and frequency scaling (DVFS) (by selecting the performance states or p-states of the tasks to manage the power and thus the lifetime of the processor running different tasks over their periods. We considered both single-rate and multi-rate embedded systems with preemption. We framed the reliability optimization problem as the continuous constrained nonlinear optimization problem in which the system EM-induced reliability is maximized subject to the timing constraints, which is further solved by simulated annealing method. Experimental results have shown that for low utilization systems, significant reliability improvement can be achieved with even smaller power consumption than existing reliability-ignore scheduling method. The proposed can lead to near Pareto's front trade-off between the power/energy and the lifetime compared to the existing task scheduling method.

References

[1] "Critical Reliability Challenges for The International Technology Roadmap for Semiconductors (ITRS)," in International Sematech Technology Transfer Document 03024377A-TR, 2003.

[2] H. Aydin, R. Melhem, D. Mosse, and P. Mejia-Alvarez, "Power-aware scheduling for periodic real-time tasks," *Computers, IEEE Transactions on*, vol. 53, no. 5, pp. 584–600, May 2004.

[3] G. Buttazzo, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, 2nd ed. Springer Publishing Company, Incorporated, 2010.

[4] S. Chatterjee, M. Fawaz, and N. F. Najm, "Redundancy-Aware Electromigration Checking for Mesh Power Grids," in *IEEE International Conference On Computer-Aided Design (ICCAD)*, 2013.

[5] J.-J. Chen and T.-W. Kuo, "Procrastination determination for periodic real-time tasks in leakage-aware dynamic voltage scaling systems." in *Computer-Aided Design, 2007. ICCAD 2007. IEEE/ACM International Conference on*, Nov 2007, pp. 289–294.

[6] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba, "Advanced configuration and power interface specification 5.0a," 2013, http://www.acpi.info.

[7] L. Huang, F. Yuan, and Q. Xu, "On Task Allocation and Scheduling for Lifetime Extension of Platform-Based MPSoC Designs," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 22, no. 12, pp. 2088–2099, 2011.

[8] X. Huang, T. Yu, V. Sukharev, and S. X.-D. Tan, "Physics-based electromigration assessment for power grid networks," in *Proc. Design Automation Conf. (DAC)*, June 2014.

[9] L. Ingber, "Adaptive simulated annealing (asa): Lessons learned," *Control and cybernetics*, vol. 25, pp. 33–54, 1996.

[10] "Enhanced intel speedstep technology for the intel pentium m processor," Intel Co., 2004, http://download.intel.com/design/network/papers/30117401.pdf.

[11] "International technology roadmap for semiconductors (ITRS), 2012 update," 2012, http://public.itrs.net.

[12] M. A. Korhonen, P. Borgesen, K. N. Tu, and C. Y. Li, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, 1993.

[13] K.Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *International Symposium on Computer Architecture*, 2003, pp. 2–13.

[14] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard-real-time environment," *J. ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973. [Online]. Available: http://doi.acm.org/10.1145/321738.321743

[15] Z. Lu, W. Huang, J. Lach, M. Stan, and K. Skadron, "Interconnect lifetime prediction under dynamic stress for reliability-aware design," in *Proc. IEEE/ACM International Conferene on Computer-Aided Design (ICCAD)*. IEEE, Nov. 2004, pp. 327–334.

[16] R. Melhem, D. Mosse, and E. Elnozahy, "The interplay of power management and fault recovery in real-time systems," *Computers, IEEE Transactions on*, vol. 53, no. 2, pp. 217–231, Feb 2004.

[17] C. Scordino and G. Lipari, "A resource reservation algorithm for power-aware scheduling of periodic and aperiodic real-time tasks," *Computers, IEEE Transactions on*, vol. 55, no. 12, pp. 1509–1522, Dec 2006.

[18] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The case for lifetime reliability-aware microprocessors," in *Computer Architecture, 2004. Proceedings. 31st Annual International Symposium on*, 2004, pp. 276–287.

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

[20] V. Sukharev, A. Kteyan, E. Zschech, and W. D. Nix, "Microstructure Effect on EM-Induced Degradations in Dual Inlaid Copper Interconnects," *IEEE Transactions on Device and Materials Reliability*, vol. 9, no. 1, pp. 87–97, 2009.

[21] V. Sukharev, "Beyond Black's Equation Full-Chip EM/SM Assessness In 3D IC Stack," pp. 1–29, Apr. 2013.

[22] O. Unsal, I. Koren, and C. Krishna, "Towards energy-aware software-based fault tolerance in real-time systems," in *Low Power Electronics and Design, 2002. ISLPED '02. Proceedings of the 2002 International Symposium on*, 2002, pp. 124–129.