

Energy and Lifetime Optimizations for Dark Silicon Manycore Microprocessor Considering Both Hard and Soft Errors

Taeyoung Kim, *Student Member, IEEE*, Zeyu Sun, *Student Member, IEEE*, Hai-Bao Chen, *Member, IEEE*, Hai Wang, and Sheldon X.-D. Tan, *Senior Member, IEEE*

Abstract—In this paper, we propose a new energy and lifetime optimization techniques for emerging dark silicon manycore microprocessors considering both hard long-term reliability effects (hard errors) and transient soft errors, which have been studied less in the past. We consider a recently proposed physics-based electromigration (EM) reliability model to predict the EM-induced reliability. We employ both dynamic voltage and frequency scaling (DVFS) and dark silicon core state using ON/OFF switching action as the two control knobs. We show that on-chip power consumption has different (even contradicting) impacts on soft and hard reliability effects. This paper also shows that soft error should be mitigated by other techniques if aggressive low power and high long-term reliability are pursued. We focus on two optimization techniques for improving lifetime and reducing energy. To optimize EM-induced lifetime, we first apply the adaptive Q-learning-based method, which is suitable for dynamic runtime operation as it can provide cost-effective yet good solutions. The second lifetime optimization approach is the mixed-integer linear programming (MILP) method, which typically yields better solutions but at higher computational costs. To optimize the energy of a dark silicon chip subject to the both hard and soft reliability effects, power budgets, and performance limits, the Q-learning method has been applied as well. A large class of multithreaded applications is used as our benchmarks to validate and compare the proposed dynamic reliability management methods. Experimental results on a 64-core dark silicon chip show that the proposed DRM algorithm can effectively manage and optimize the lifetime of a dark silicon microprocessor under the given power budget and performance limit. Also, the proposed energy optimization can effectively manage and optimize energy consumption subject to both hard and soft-error rates, power budget, and performance limits as constraints. We also show that the under tightened power and performance constraints, we cannot satisfy both hard and soft errors at the same time as there is no simple

tradeoff between performance/power and reliability in this case. Some other soft-error mitigation techniques are required in this case.

Index Terms—Dark Silicon, electromigration, integrated circuit reliability, microprocessors, optimization, Q-learning, reinforcement learning, reliability, semiconductor device reliability.

I. INTRODUCTION

FOR the last several decades, technology scaling has led to the continuous integration of devices, and microprocessors will have more cores integrated in the future. However, due to the failure of Dennard's scaling [1], chip power density is increasing on technology nodes since transistor and voltage scaling is no longer linear. The consequence is the emergence of so-called dark silicon manycore microprocessors, which mean only a percentage of cores can be powered on the chip due to the power and temperature limitations. Recently, architecture researchers predicted that future many-core (100–1000 cores) silicon dies can only be powered up partially (so-called dark silicon) as power constraints will not allow all the cores to be active at the same time. Such manycore systems pose new challenges and opportunities for power/thermal and reliability management of those chips [2].

Existing works for dark silicon research have been focused mainly on the core organization, optimal number of cores, task allocation, migration, and scheduling [2]–[5]. Moreover, those existing works focus on performance latency, bandwidth, and energy efficiency for dark silicon chips. Recently, the reliability management methods for dark silicon manycore scaling have been studied [6], [7]. However, all of these works considered general reliability models, which will not be accurate for specific failure mechanisms. Recently, a new electromigration (EM) model has been used for energy optimization as a dynamic reliability management (DRM) but it only considered the EM model [8]. For dynamic power and thermal management, learning-based methods have recently become popular. Many proposed methods applied Q-learning-based method, which is a reinforcement machine learning method for the adaptive control [9]–[12].

Energy-efficient or green computing is important for sustainability and environmental responsibility. This is also true for dark silicon many-core microprocessors as they may power many IT equipment and data centers in the near future. Power, performance, and temperature limitations are traditional dominant factors for energy efficient high-performance

Manuscript received August 30, 2016; revised January 1, 2017 and April 8, 2017; accepted May 9, 2017. Date of publication June 13, 2017; date of current version August 23, 2017. This work was supported in part by NSF under Grant CCF-1527324 and in part by NSFC under Grant 61404024 and Grant 61604095. (*Corresponding author: Taeyoung Kim.*)

T. Kim is with the Department of Computer Science and Engineering, University of California, Riverside, CA 92521 USA (e-mail: tkim049@cs.ucr.edu).

Z. Sun and S. X.-D. Tan are with the Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521 USA.

H.-B. Chen is with the Department of Micro/Nano-Electronics Engineering, Shanghai Jiao Tong University, Shanghai 200240, China.

H. Wang is with the School of Microelectronics and Solid-State Electronics, University of Electronics Science Technology of China, Chengdu 610051, China.

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVLSI.2017.2707401

and mobile computing. As technology advances, reliability starts to become another limiting factor in high-performance nanometer microprocessors due to the high failure rates in deep submicrometer and nanoscale devices. It is expected that future chips will show signs of reliability-induced aging much sooner than the previous generations based on the prediction of ITRS 2014 [13].

Among many reliability effects, we consider EM and soft error-induced reliability effects as they have become major concerns for designers due to aggressive transistor scaling and increasing power density. EM effect is the dominant interconnect failure mechanism in the 22 nm and below technology due to the shrinking wire width and thermal elevation due to FinFET devices [14], which will have immediate impacts on the metals above the FinFET devices. We want to stress that there exist many other long-term reliability effects such as negative-bias temperature instability (NBTI), hot carrier injection (HCI), and time dependent dielectric breakdown (TDDB) for devices and stress migration and thermal migration for interconnects. However, in this paper, we only consider EM reliability for the demonstration of the proposed reliability management techniques. The proposed techniques are orthogonal to other long-term reliability managements as those long-term reliability effects generally behave similarly or in a similar trend under their stressing conditions in terms of voltage, current, and temperature [15].

On the other hand, soft-error related reliability has quite different impacts on VLSI chips (from the long-term reliability). This is especially true for chips operating in the very low voltage or even near threshold voltage regions. For practical chips, we have to consider both reliability effects at the same time. Although there are many soft-error mitigation techniques ranging from redundancy-based design to software-based methods, it is important to study their impacts in the context of long-term reliability optimization techniques such as dynamic voltage and frequency scaling (DVFS) and ON/OFF switching of cores from dark silicons.

As a result, in this paper, we consider the energy or reliability optimization subject to the two kinds of reliability (long term and soft errors) and the power, performances, and temperature constraints. We look at the two system-level control knobs: core status knobs to enable or disable a core due to dark silicon requirement and DVFS knobs for traditional power and thermal management. Hence, in this paper, we try to solve the resulting optimal control problems to seek the best policies for DVFS and core status in the context of the two kinds of reliability constraints.

In this paper, we propose a new energy and lifetime optimization techniques for emerging dark silicon manycore microprocessors considering both hard long-term reliability effects (hard errors) and transient soft errors. Our new contributions lie in the following aspects.

- 1) We consider a recently proposed physics-based EM reliability model to predict the EM-induced long-term reliability. For system-level soft error modeling, we consider the DVFS-aware soft error rate (SER) model and the sum of failure rates (SOFR) method are employed.

- 2) To model dark silicon, we consider thermal design power as the power constraint for dark silicon manycore microprocessors. We employ both DVFS and dark silicon core state using ON/OFF switching action as the two control knobs. We show that on-chip power consumption has different (even contradicting) impacts on soft and hard reliability effects. This paper also shows that soft error should be mitigated by other techniques if aggressive low power and high long-term reliability are pursued.
- 3) We focus on two optimization techniques for improving lifetime and reducing energy. To optimize EM-induced lifetime, we first apply the adaptive Q-learning-based method, which is suitable for dynamic runtime operation as it can provide cost-effective yet good solutions. The second lifetime optimization approach is the mixed-integer linear programming (MILP) method, which typically yields better solutions but at higher computational costs.
- 4) To optimize the energy of a dark silicon chip subject to the both hard and soft reliability effects and performance constraints, the Q-learning method has been applied as well. A large class of multithreaded applications is used as the benchmark to validate and compare the proposed DRM methods.
- 5) Experimental results on a 64-core dark silicon chip show that the proposed DRM algorithm can effectively manage and optimize the lifetime of a dark silicon microprocessor under the given power budget and performance limits. Also, the proposed energy optimization can effectively manage and optimize energy subject to both hard and SERs, given power budget and performance limits as constraints. We also show that with the tightened power and performance constraints, we cannot satisfy both hard and soft errors at the same time as there is no simple tradeoff between performance/power and reliability in this case. Some other soft-error mitigation techniques become imperative in this case.

This paper is organized as follows. Section II reviews recently proposed EM models, system-level EM modeling and soft-error models at the system level. Section III introduces the new DRM methods considering both EM-induced hard reliability and soft errors. We present two learning-based optimization techniques, one for optimizing energy and another for optimizing lifetime subject to performances, temperature, and reliability constraints. Section IV presents some numerical results on a simulated dark-silicon framework. Section V concludes this paper.

II. REVIEW OF THE EM AND SOFT ERROR RELIABILITY MODELS

A. New Physics-Based EM Modeling and Analysis

EM is the 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.

The EM effects are mainly modeled and heavily used by empirical Black's equation [16] and Blech limit [17]. However, those models are not physics-based and they do not fully consider the predictability for varying stressed conditions and complicated interconnect wire structures. In addition, they do not address the inherent redundancy in the power grid networks that are the most vulnerable wires in a chip.

To address those problems, a more physics-based compact EM model has been recently proposed for full-chip reliability analysis [18], [19], which is the basis for EM reliability modeled in this paper. 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, and the residual stress of the wire due to thermal and other effects as well as other wire geometry and material parameters. The approximate value of void nucleation time (t_{nuc}) is determined as the instant in time when stress at the cathode end of the line reaches σ_{crit} , which 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 [20]) in the confined 1-D line

$$t_{\text{nuc}} \approx \tau^* e^{\frac{E_V}{kT}} e^{-\frac{f\Omega}{kT} \left(\sigma_{\text{Res}} + \frac{eZ\rho l}{4\Omega} j \right)} \times \ln \left\{ \frac{\frac{eZ\rho l}{4\Omega} j}{\sigma_{\text{Res}} + \frac{eZ\rho l}{4\Omega} j - \sigma_{\text{CR}}} \right\} \quad (1)$$

where $\tau^* = (l^2/D_0)e^{E_D/kT} (kT/\Omega B)$. 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 energies 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), and 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 $t > t_{\text{nuc}}$. The wire resistance starts to increase over the time in the growth phase. Since the drift velocity of the void edge relates to atomic flux as $\vartheta = \Omega j$ [21], we can express it as $\vartheta = (D_a/k_B T)eZ\rho j$. Kinetics of the line resistance change can be approximately described as

$$\Delta r(t) = \vartheta(t - t_{\text{nuc}}) \left[\frac{\rho_{Ta}}{h_{Ta}} \left(\frac{1}{2H} + \frac{1}{W} \right) - \frac{\rho_{Cu}}{HW} \right]. \quad (2)$$

Here, ρ_{Ta} and ρ_{Cu} are the resistivity of the barrier material (Ta/TaN) and copper, W is the linewidth, H is the copper thickness, and h_{Ta} is the barrier layer thickness. As a result, the p/g network becomes a time-varying network and its voltage drops will keep changing over the time [18].

1) *EM Assessment at Power Grid Level*: Because of the concern with the long-term average effects of the current, in EM related work, such as [22], a dc model of the power grid is generally assumed. At the chip level, EM failure typically is defined not by the failure of a single wire, but

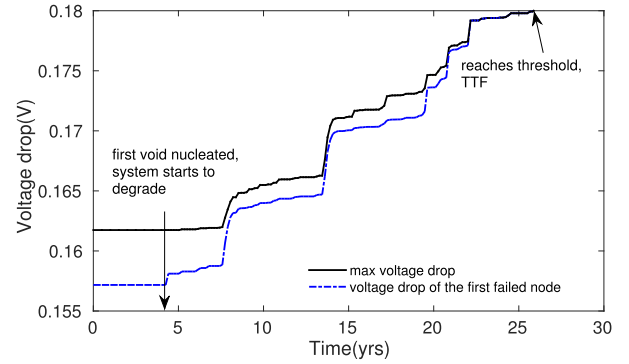


Fig. 1. Voltage of the first failed node in different simulation time.

by the voltage drop of a power grid network as power grid network is most vulnerable to EM failures. In our problem formulation, each mortal wire, which is subject to the EM effect, will start to change its resistance value upon achieving the nucleation time. Finally, we end up with the power grid systems, which is linear, time-varying, and driven by the dc effective currents, which is modeled as $G(t)v(t) = I_{\text{eff}}$, where $G(t)$ is 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 of unknown voltages. Because EM is a long term effect, the time scale used in this paper is measured in months or years. In the EM-induced reliability analysis at a power/ground level, the voltage drops of the grid can be computed at the fixed time step for each core. Specifically, the resistance of one or more wires begins to increase after the nucleation time defined in (1) and then the resistance of a wire will start to increase based on (2). At each time step, we collect new wires whose nucleation times were reached and compute the new resistance for existing wires in the growth phase and the corresponding voltage drops of the whole grid. This process is repeated until the voltage drop of one or more nodes exceeds the critical voltage drops allowed (defined as 10% of V_{DD}) [23].

Fig. 1 shows one example in which the voltage of one node in a p/g network varies time after the creation of the first void in the network and its value can be tracked [23].

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 (defined as 10% of V_{DD}). In our dark silicon problem, we use the same mesh-structured p/g network for all the cores.

2) *System-Level EM Reliability Model*: At the system-level EM reliability, the manycore system will run on different tasks under different p-states.

Let us assume that we have a set of different time intervals Δp_k characterized by different workloads or p-state in terms of current density j_k and temperature T_k for a processor or a core. It means that $P = \sum_{k=1}^n \Delta p_k$ is the total execution time. Each

k th workload, if it lasts till imaginary failure provides time to failure TTF_k . Thus, the failure rate at the k th workload, which lasts Δp_k is $\lambda_k = 1/TTF_k$. Then the average failure rate for the considered set of workloads is

$$\lambda_{\text{avg}} = \sum_{k=1}^n \frac{\Delta p_k}{\sum_{j=1}^n \Delta p_j} \lambda_k = \sum_{k=1}^n \frac{\Delta p_k}{P} \lambda_k. \quad (3)$$

As a result, the expected time to failure (TTF) or average lifetime of the whole processor, TTF is [24]

$$\text{MTTF} = \frac{1}{\lambda_{\text{avg}}} = \frac{1}{\left(\sum_{k=1}^n \left(\Delta p_k \frac{1}{\text{MTTF}_{R,k}} \right) \right) / P} \quad (4)$$

where $\text{MTTF}_{R,k}$ is the actual Mean TTF (MTTF) under the k th power and temperature settings for Δp_k period, assuming the chip works through n different power and temperature settings and $P = \sum_{k=1}^n \Delta p_k$. Each $\text{MTTF}_{R,k}$ will be computed based on the EM models discussed in Section I.

To consider a system-level EM reliability on a manycore dark silicon processor, we use the shortest lifetime among all the cores as the lifetime for all manycore processors [25]. We want to stress that the proposed techniques are orthogonal to other long-term reliability effects (such as NBTI, HCI, and TDDDB for devices and stress and thermal migration for interconnects). The proposed techniques are orthogonal to other long-term reliability managements as those long-term reliability effects generally behave similarly or in a similar trend under their stressing conditions in terms of voltage, current, and temperature [15]. Specifically, the power and temperature typically have the same impacts on the NBTI, HCI, and TDDDB as those failure effects follow the Arrhenius equation for the relationship between the failure rate and temperature (which is a function of powers or energy) [26], as a result, those long-term reliability effects will become worse when temperature increases. As a result, DVFS-based optimization will lead to similar tradeoff between long-term reliability and soft errors.

B. Soft-Error Reliability Model Considering DVFS Impacts

Soft errors, or single event upset (SEU), are defined as the transient faults inside the logic or memory on a chip, and result in an incorrect system output. The soft errors can be caused by cosmic radiation, alpha particle decay, and thermal neutrons. SER is the rate in which a chip or system encounters soft errors and typically can be expressed as the number of failures in the given time. Although there is still a lack of consensus on the exact SER of specific chips and systems, it is obvious that the SER per chip is practically increasing due to the increasing number of components or cores on a chip. Recently it has been reported that the DVFS method used for energy saving, negatively affects the system reliability because the transient fault rate increases and the critical charge decreases by lowering the voltage and frequency. As a result, new exponential soft error models have been introduced to account for those effects [27], [28].

For our problem, we employ an existing exponential model considering DVFS effects on SER, which assumes that the radiation-induced failure follows a Poisson distribution, so the

average SER can be expressed as terms of operating frequency f and supply voltage V_{dd} , where SER_0 is the average failure rate at the maximum frequency f_{max} and voltage V_{max} , (so, $f_{\text{min}} < f < f_{\text{max}}$, $V_{\text{min}} < V < V_{\text{max}}$) in (5) [27]

$$\text{SER}(f, V_{\text{dd}}) = \text{SER}_0 e^{\frac{d(f_{\text{max}}-f)}{f_{\text{max}}-f_{\text{min}}}} \quad (5)$$

where d is an architecture dependent constant, which is the sensitivity of failure rate with DVFS. We also employ the previous work to model the relationship between operating frequency and supply voltage to further simplify (5) from [29]

$$f = \beta \frac{(V_{\text{dd}} - V_{\text{th}})^2}{V_{\text{dd}}} \quad (6)$$

where β is a technology-related constant and V_{th} is the threshold voltage. By substituting (6) into (5), DVFS-aware SER equation can be derived as the function of only supply voltage V_{dd} only [28]

$$\text{SER}(V_{\text{dd}}) = \text{SER}_0 e^{\frac{d(f_{\text{max}} - \beta V_{\text{dd}} - 2V_{\text{th}} + \frac{V_{\text{th}}^2}{V_{\text{dd}}})}{f_{\text{max}} - f_{\text{min}}}}. \quad (7)$$

1) *System-Level Soft Error Model for Dark Silicon Many-core Processor*: To estimate system-level soft error reliability, SOFR with architecture vulnerability factor (AVF) method has been widely accepted in the semiconductor industry [30], [31] for combining SERs from each core to estimate whole system-level soft error. The AVF is used to express the probability that a visible soft error will occur with a given raw error event in a core [32]. The previous study shows that the SOFR model can be used to show the whole system SER (SER_{sys}) [31]

$$\text{SER}_{\text{sys}} = \text{SOFR} = \sum_{i=0}^m \text{AVF}_i \times \text{SER}_i(V_{\text{dd}}) \quad (8)$$

where m is the total number of cores in a processor, $\text{SER}_i(V_{\text{dd}})$ is the SER for given voltage setting (V_{dd}), and AVF_i is AVF for i th core.

C. Impact of Process Technology on Soft-Error Reliability Model

In the past, soft errors in microprocessor logic have not been greatly concerned as the number of flops/latches in a microprocessor is much fewer than the number of SRAM cells and microprocessor SEU rates were lower than SRAM SEU rates. After 90-nm technology, however, microprocessor SEU rates are larger than SRAM SEU rates because flop protection mechanisms (machine encoding and invariant checking) are more difficult to implement than simple memory protection mechanisms (parity, error correction code) [33]. Thus, the SEU in microprocessor will be the dominant factor to system SER as technology scales to a smaller feature size. Table I shows that the normalized SEU rate in microprocessor reported from the real silicon data [34]. As technology scales to a smaller feature size, SEU rates keep increasing for different technologies. To assess technology scaling impacts, we used different raw SER values and other parameters [different threshold voltage (V_{th}) and supply voltage (V_{dd})] in (7) based on the experimental data from [34] and [35]. The technology scaling impact on our proposed DRM method will be discussed in Section IV-D.

where $\alpha(t)$ is learning rate between 0 and 1, which determines how much newly calculated Q-value will be applied. For instance, for α is 0, the agent is not learning anything, or for 1, the agent is always considering the most recent state-action. In practice, the constant learning rate is used ($\alpha(t) = 0.1, \forall t$) as the algorithm needs to converge, so it requires a learning rate close to zero [37]. $s(t+1)$ is determined by action $a(t)$, so $Q^t(s(t+1), a)$ are all possible actions' Q-values from future state. So the discount factor γ (between 0 and 1) affects the importance of future penalty. A small discount factor gives more penalties in the near future penalty, and high discount factor accounts more for the far future penalty. This parameter needs to be tuned experimentally. $\min(\forall Q^t s(t+1), a)$ can be viewed as the estimate of the optimal future value. The difference between old Q-value (Q^t) and learned value ($PT(t+1) + \gamma \min(\forall Q^t (s(t+1), a))$) updates the new Q-value (Q^{t+1}) with the learning rate.

The $(PT(t+1))$ in (9) at $t+1$ time, is the penalty obtained after performing action $a(t)$ in state $s(t)$ on the dark silicon manycore processor. In our problem, we have three main constraints: total core power, performance deadline of all the tasks, and temperature upper limit. EM-induced lifetime is what we want to maximize. As a result, we define the penalty function PT in [11] and [8] to consider multiple constraints.

We can build a PT as shown in (10) for each EM-induced and energy optimizations. PT_E is a PT for total core energy, PT_{EM} is a PT for EM-induced lifetime, PT_{SER} is a PT for system-level SER, PT_{power} for power, PT_{temp} for temperature, and PT_{perf} for performance deadline of all tasks. Each PT_x is normalized in (10). We use the feature scaling method to bring all values between 0 and 1. For instance, $PT_E = (E(t+1) - E(t)/E_{Max} - E_{Min})$ for energy related penalty, where $E(t)$ is the total energy consumption in the previous time t and $E(t+1)$, is energy of the system at current $t+1$. For the EM lifetime, $PT_{EM} = (MTTF(t) - MTTF(t+1)/MTTF_{Max} - MTTF_{Min})$ for EM related penalty where $MTTF(t)$ is the MTTF of the system for EM induced in the previous time t and $MTTF(t+1)$ is the MTTF of system at current $t+1$. Similarly, for the soft errors, $PT_{SER} = (SER(t+1) - SER(t)/SER_{Max} - SER_{Min})$ for soft error related penalty where $SER(t)$ is the SER of the system in the previous time t and $SER(t+1)$ is the SER of system at current $t+1$. Energy and EM terms can be interchangeable, so both energy and life optimizations can be achieved with similar PT as seen in (10)

$$PT = PT_E + C \sum_{x=\{EM, SER, power, temp, perf\}} \delta_x PT_x$$

for energy optimization

$$PT = PT_{EM} + C \sum_{x=\{E, power, temp, perf\}} \delta_x PT_x$$

for EM-induced lifetime optimization

$$\delta_x = \begin{cases} 0 & \text{if } PT_x \leq B_x + \Delta_x \\ 1 & \text{if } PT_x > B_x + \Delta_x \end{cases} \quad (10)$$

where δ_x is a binary function to active ($\delta_x = 1$) or inactive ($\delta_x = 0$) of user defined or given constraint bounds, B_{power} , B_{perf} , and B_{temp} in the PT. They are also normalized

power, performance, and temperature bounds, respectively. Each Δ_x is the difference between each bound and average penalty (PT) for power, performance, and temperature. Δ_x is negative if the system violates the given constraint, otherwise, it is positive and the system is bounded and performs well. Therefore, if the system violated the user constraints in the past, then the penalty can be quite significant [due to large value for constant C in (10)].

Our learning-based lifetime/energy optimization algorithm steps can be explained as follows: the input is an initial state set for each core with p-state and core status and output is the selected p-state and core states. First, all the Q-values in the Q-table are initialized to zero. The current state, denoted as $s(t)$, finds an action $a(t)$ with the lowest Q^t in (9) and switches to next state with corresponding p-states and active cores. For every step, EM lifetime, SER, performance, temperature, and power are evaluated, and thus, the whole environment can be updated. Then, it calculates the new corresponding penalty $PT(t+1)$ in (10) and Q^{t+1} can be updated (learning process). After the update, the current state is discarded in exchange for a new action and subsequent iterations yield more updates with new states. Finally, when all the Q-values changes are less than a certain threshold, the best policy will be chosen.

B. MILP-Based Formulation and Solution for Lifetime Optimization

The second approach for lifetime optimization that we apply is the MILP method. MILP formulation of the performed constrained lifetime optimization method is more straightforward than the Q-learning-based method. MILP also delivers better results than Q-learning-based method as shown in this paper. However, in general, MILP has higher computation costs than the Q-learning method. Also, MILP solver is very heavier, so it is not suitable for online management method. Hence, we can use MILP solution to measure the solution quality of the Q-learning-based method.

We know that the MTTF of a core (or lifetime) stressed by different periods with different temperature can be approximated by formula (4). For the manycore processor, we assume that the MTTF of the overall chip is determined by the minimum MTTF of all cores [25]. Let us define the lifetime of a core k for a given state s_i as $L(k, s_i)$, which will be built as a look-up first. Then, the lifetime optimization problem for dark silicon manycore processor can be formulated as the following MILP problem:

$$\begin{aligned} \max_i \min_k & \sum_i \sum_k b_i c_k L(k, s_i) \\ \text{s.t. } \forall i & : \sum_i b_i = 1, \quad \forall k: \sum_k c_k = 1 \\ & \sum_k Power_k \leq B_{power} \\ & Temp_{max} \leq B_{temp}, \quad Perf_{max} \leq B_{perf} \end{aligned} \quad (11)$$

where $Power_k$ is k th normalized core power, $Temp_{max}$ is maximum normalized temperature among cores, and $Perf_{max}$ is maximum normalized performance deadlines among cores

(or all the tasks). B_{const} , B_{const} , and B_{const} are normalized performance, temperature, and power bounds allowed. Note that a selection for a chip lifetime can be denoted by a Boolean variable b_k , which equals to 1 if the k th core's lifetime is selected and 0 otherwise. Similarly, a state selection for a core is also denoted by a Boolean variable c_i , which equals to 1 if state s_i is selected for the chip and 0 otherwise.

C. Implementation of the Dark Silicon Evaluation Platform

To evaluate the proposed DRM algorithms, we implement a simulation-based platform for dark silicon processor. The platform is shown in Fig. 2. We first describe the major component models of the framework such as microarchitecture, power estimation, and thermal and reliability models. Our proposed framework uses Sniper as an microarchitecture model, which is an accurate and fast application-level interval-based microarchitecture simulation [39]. The interval simulation is a recently proposed multi/manycore simulation framework at a higher level of abstraction, which is faster than cycle-accurate full-system simulation. The interval simulation uses mechanistic analytical model, which is constructed from the mechanism of a superscalar processor core. The cycle-accurate full-system simulator, such as gem5 (full-system mode) [40], GEMS [41], MARSSx86 [42], and SimFlex [43] can run both application and operating system (OS). These frameworks have the merit of having an accurate evaluation of I/O activities and OS extensive kernel function. However, these full-time simulations are extremely slow and not very suitable for our framework because they rely on the existing OS systems, which currently do not support manycore and dark silicon architectures in their simulators [44]. Thus, to support dark silicon and many-core processor, we choose application-level Sniper simulator. This Sniper interval-based model is accurately matching well with the Intel x86 multicore architecture [39]. PARSEC [45] and SPLASH-2 [46] benchmarks are used for our platform workloads. PARSEC benchmark is recently released multithreaded benchmarks, which provide an up-to-date collection of modern workloads for multi/manycore systems, and SPLASH-2 has been used many multi/manycore research for a long time. We use both workloads to evaluate our proposed framework and algorithm in Section IV.

For the power estimation, we use McPAT (multicore power, area, and timing), which is a recently proposed full integration modeling framework. McPAT can provide dynamic and static, even short-circuit power dissipation, and provides multithreaded and multicore processor models. For the thermal model, we use HotSpot to accurately characterize the thermal traces from the given multithreaded task run in each core [47]. To enable the dark silicon feature, the floor plan and power trace are dynamically controlled by the dark silicon DRM module in Fig. 2.

As shown in Fig. 2, once the cycles per instruction stacks and power/energy traces are achieved in the microarchitecture model with the power model, the thermal model can generate thermal traces for a given task run. With each core's power trace, thermal trace, core voltage, core frequency, and active cores, we can perform EM and soft-error reliability effects

analysis and the system-level assessment for microprocessor lifetime-based on the reliability models.

We remark that the soft error affects the short term hardware functionality and it has a different way to impact the reliability of the circuits than the long-term reliability. However, both effects hurt the reliability of a chip, and we think it is necessary to consider both as the power/energy and performance have contradicting impacts on them. The tradeoff has to be found among the robustness, costs, and performance of the many-core processor systems to mitigate the both soft and hard reliability effects.

In our formulation, both soft errors and EM-induced reliability are modeled in terms of system SER and MTTF, respectively, and parameterized by the chip and system parameters such as V_{DD} /frequency, temperature, p-state, and so on. As shown in (9) and (10), the SER and MTTF will contribute the constraints of the optimization in terms of PTs.

As seen in Fig. 2, our framework implemented two reliability models, such as EM [18] for long-term reliability and DVFS-aware soft-error effects [27], [28], [31] for short-term reliability. Each reliability has been assessed for DRM module to provide constraints in our Q-learning.

We also stress that DVFS and task scheduling may not be the most effective way to mitigate the soft errors, and other techniques are required when the soft errors are high due to the conflicting requirement from hard reliability.

D. Time Complexity Analysis

It has been proven that MILP problems are NP-hard. Though branch and bound techniques can be used to solve the problem, the time complexity is not easy to analyze as we use commercial CPLEX as the solver. For the Q-learning, each value iteration can be performed in $O(|A||S|^2)$ steps, or faster if there is sparsity in the transition function (where A is a number of actions and S is a number of states). In practice, policy iteration converges in fewer iterations than value iteration, and there is no known tight worst case bound available [48]. As a result, we report the running CPU times of Q-learning and MILP in our experiments with more p-states in Table III in Section IV to compare the time complexities of the two methods. We further remark that it is rather difficult to fairly compare CPU times for these two optimization methods as the Q-learning was running on Python, whereas the MILP was running on commercial CPLEX optimizer. We only measure solving overhead time for Q-learning and MILP. Nevertheless, the numerical results still show that MILP has much higher computation cost than the Q-learning method.

E. Practical Application of the Proposed DRM With Reliability Models

Currently, it is very hard to build on-chip EM sensor to directly measure time to EM failure of a core for its current temperature and power consumption (although we have made some early efforts on on-chip EM sensors recently [49]). But at the full chip level, as far as power grid is concerned, EM can be measured by the voltage drops in power grids as

we discussed in Section II. In our many-core dark silicon simulation framework, the EM-induced TTF assessment technique at circuit level discussed in this paper (which was proposed in [18] and [23]) was used for each core.

For practical application of proposed reliability management method for a real chip, such EM and SER assessment techniques need to know the details of the power grid networks and power consumptions of gates or function modules for some real workloads. As a result, it can be built when the chip was designed and then power grid voltage drop sensors (for EM measurement) or lookup table/other behavior models can be constructed for TTF as the function of temperature and power inputs, which can be measured or estimated accurately. The accuracy of the reliability assessments with respect to the real silicon data needs to be calibrated in the accelerated testing conditions for real chip under practical workloads, which will go beyond the scope of this paper and can be future research. We further stress that the on-chip temperature and the powers, which actually can be measured or accurately predicted. For instance, Intel's multicore CPU has the one thermal sensor per core [50]. The power or functional block of the cores can be measured or estimated accurately using the performance counters [51]. Then, our models proposed in this paper can be applied to core-level reliability management.

We notice that the learning-based DVFS managements have been used before to deal with difficulty of controlling the dynamics of the multi/manycore processors [52], [53]. Recently, reinforcement learning has been successfully applied for DVFS management of multi/many-core systems [7], [9], [54]. These approaches employ a simple type of reinforcement learning (Q-learning) because this method has a relatively low overhead in terms of execution time and memory foot print. In this paper, we assume the similar light scheduling overheads or time costs by the Q-learning methods. We have added the execution time for the proposed Q-learning method in Section IV-E.

Furthermore, as the state space of many-core systems become large, we can explore overhead-aware scheduling method [54], the discrete lookup table method [55], [56], or pretraining-based method [7] during the Q-learning to avoid expensive Q-value updating operations. In this paper, we use lookup table-based method.

IV. NUMERICAL RESULT AND DISCUSSIONS

A. Experimental Setup

The proposed new energy optimization algorithm in the dark silicon framework has been implemented in Python 2.7.9 with the numerical libraries (Numpy 1.9.2 and Scipy 0.15.1). For dark silicon framework, we modified the architectural simulator (Sniper 6.1), power estimator (McPAT 1.0.32), and thermal simulator (HotSpot 5.02 [47]) to estimate EM-induced lifetime and system-level SER on top of the new physics-based EM model [18] and DVFS-aware soft error model [27], [28], [31]. In the proposed framework as shown in Fig 2, each simulator module is connected with a plugin connector, so that one simulator's result can dynamically feed the other's inputs. The learning agent and Q-learning method have been implemented

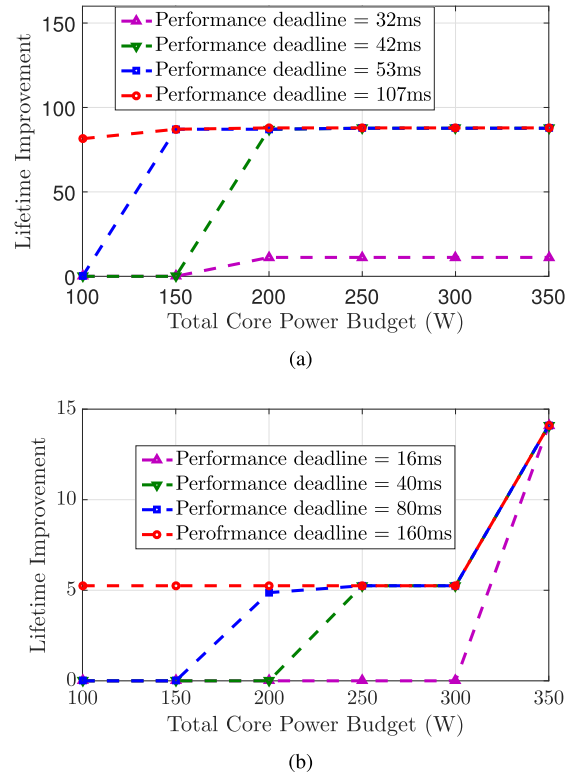


Fig. 3. Lifetime improvements given power budget and performance deadline on 64-core dark silicon chip. (a) PARSEC small task set. (b) SPLASH-2 large task set.

in Python 2.7.9 with extensive solutions from Python Numpy Extension.

Our energy optimization method is validated with a 64-core processor on the PARSEC and SPLASH-2 multithreaded benchmarks. A small number of tasks with PARSEC (1 BLACKSCHOLES, 1 CANNEAL, 1 FREQMINE, and 1 VIPS), and for a large number of tasks with SPLASH-2 (16 CHOLESKYs, 16 RADIXs, 16 RAYTRACEs, and 16 VOLRENDs) are used with the same 64 threads.

We chose two p-states with the clustered DVFS [57], which have been employed to reduce the simulation time with small solution quality degradation due to the large number of cores in our experiment. To show that our method can find the lowest possible energy optimization, we compare our results with the global DVFS method, in which all active cores have the same p-state.

The full power mode (2.0 GHz, 1.2 V) and the low power mode (1.0 GHz, 0.9 V) have been set for our framework. For the soft error model, we use system-level SERs, the architecture constant ($\beta = 1.5 \times 10^{10}$ and $d = 2$), AVF (a constant = 0.5) for each core, and the threshold voltage ($V_{th} = 0.9$) have been obtained from [31] and Enhanced Intel Speedstep Technology [58] with 45-nm technology.

B. Evaluation of the Proposed Q-Learning Lifetime Optimization Method

First, we evaluate our learning-based DRM method (see Section III) by showing lifetime improvements with different sets of power budgets and performance deadline. Fig. 3

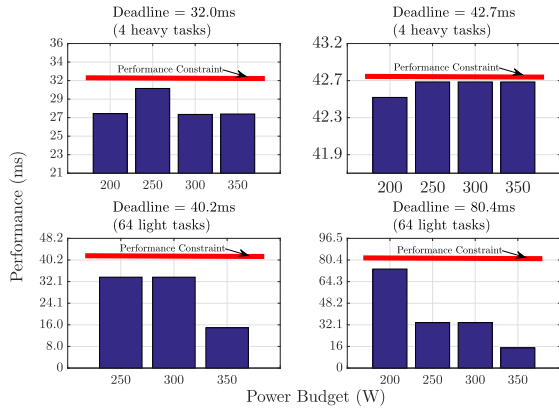


Fig. 4. Q-learning result for performance deadline from (a), (b) PARSEC tasks and light and (c), (d) SPLASH-2 tasks in 64-core dark silicon chip.

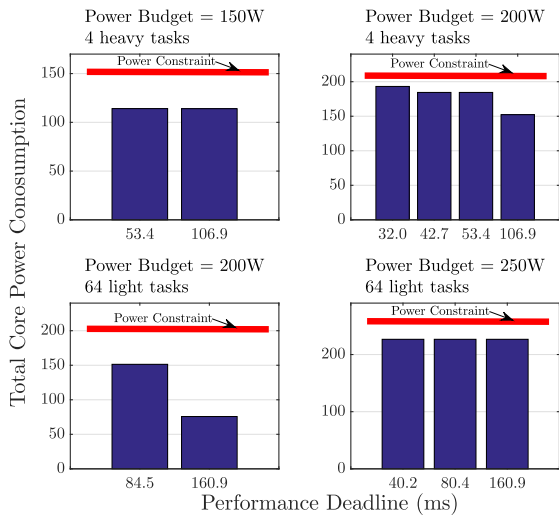


Fig. 5. Q-learning result for power constraints from heavy 4 PARSEC tasks and light 64 SPLASH-2 tasks in 64-core dark silicon chip.

shows the lifetime improvements given power budget and performance deadline for a small and large task set on 64-core dark silicon chip. As we can see in Fig. 3(a), for the small task set case, our method finds relatively high lifetime improvement (87.9 years) as the task loads are small and more cores can be in low power mode or turned OFF (dark silicon) with the given power budget and performance deadline. In small performance deadline (42 ms), there is still chance to improve lifetime (11.2 years) in the high power budgets (200–350 W). However, for the large task set case, lifetime improvement will be limited as shown in Fig. 3(b). The highest lifetime improvement is 28 years with highest power budget, and there are still 10.5 years lifetime improvement in the middle range of power budget and performance deadline (40–80 ms). This indicates that significant improvement can be made for both small and large tasks with a given power budget and performance deadline.

Figs. 4 and 5 show the power consumptions and performances from our proposed DRM method, and it indicates all the results can meet the given power budgets and performance deadline. Furthermore, no violations were found in either small or large task set results.

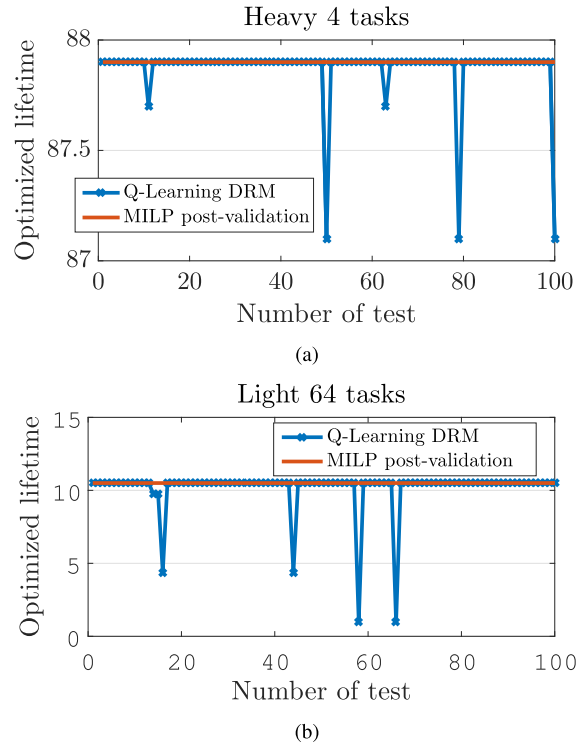


Fig. 6. Postvalidation with MILP for Q-learning accuracy. (a) 4 heavy task PARSEC workloads. (b) 64 light tasks SPLASH-2 workloads.

C. Accuracy and Convergence Rate of Proposed Q-Learning DRM Method

Now, we show some results from our second method for lifetime optimization method, MILP solver. To see the accuracy, we use MILP formulation (11) with the given solver, which is limited to postvalidation as MILP method is very expensive to solve for large-scale problem. Nevertheless, it can be used to show the quality of the solution obtained from the learning-based DRM method.

A comparison of the Q-learning DRM method and the MILP method show lifetime improvement for both small and large cases with results shown in Fig. 6. MILP method can deliver better results but with higher computational costs for large scale optimization. To see the accuracy comparison, 100 iterative tests are carried out for each case. For small and large cases in Fig. 6, our proposed Q-learning DRM method can achieve relatively high accuracy, 95% and 94%, respectively. It also shows that system violation can be effectively prevented by our proposed penalty function (10).

D. Hard and Soft Errors in Dark Silicon Manycore Processor

For the both reliability effects, we show the different impacts that power consumption has on both EM and soft error related reliability effects. We try to observe two reliability effects (EM-induced lifetime and system-level SER) on the 64-core dark silicon manycore processor with the different task sets. Fig. 7 shows how two failure rates ($1/MTTF_{EM}$ and SER_{sys}) change over power determined by different DVFS settings when 12 cores are turned on with a 64 thread multithreaded benchmarks. As we can see, for long-term hard

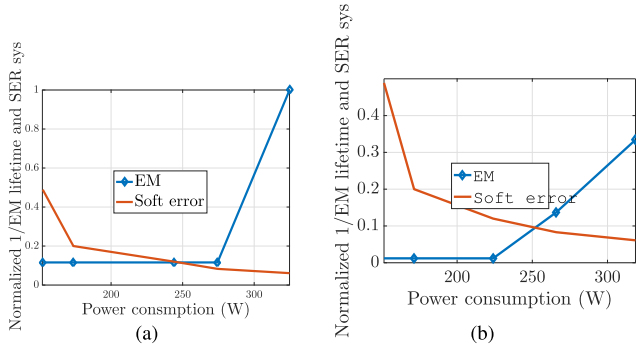


Fig. 7. Comparison between EM-induced lifetime and system-level soft error rate at different powers (by different DVFS configurations) on (a) PARSEC small tasks and (b) SPLASH-2 tasks.

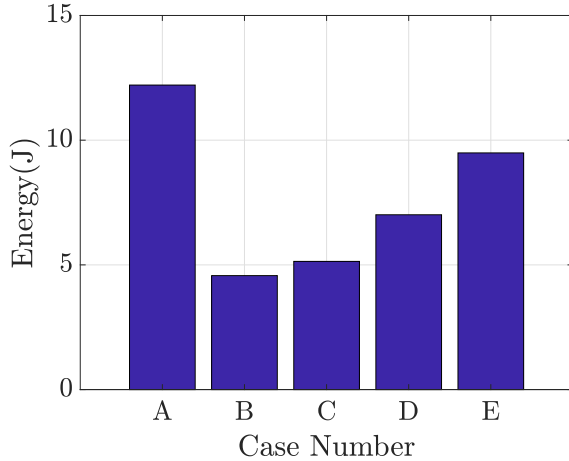


Fig. 8. Impact of different process technologies on system-level SER, from left bar, case A) Global DVFS, case B) our proposed DRM with only EM constraint, our proposed DRMs with both EM and SER constraints of case C) 45 nm, case D) 32 nm, and case E) 22 nm.

reliability (EM effect), high power leads to short lifetime. However, for short-term/transient soft errors, low power will lead to much worse reliability issues. As a result, the system level optimization subject to both reliability constraints is no longer simple tradeoff between performance/power versus reliability effects.

Fig. 8 shows that our DRM method in the different process technologies can affect to our soft error reliability model. Also, as we discussed in Section II-C, technology scaling can be impacted on SER and its constraint. Smaller technology has higher SER. As we can see in Fig. 8, with smaller technology nodes, our DRM method under loose EM (5 years) and loose SER (0.6) constraints finds less energy saving as smaller technology can affect higher SER for 32- and 22-nm cases (case D and E) than 45-nm cases (case C). However, our method still can find better energy saving point.

E. Evaluation of Proposed Q-Learning-Based Energy Optimization Method

To evaluate the proposed Q-learning-based energy optimization method in Section III, we show the total energy consumption with the different sets of EM-induced lifetime constraint, system-level SER constraint, power budgets, and performance deadlines. Energy optimization results for a small

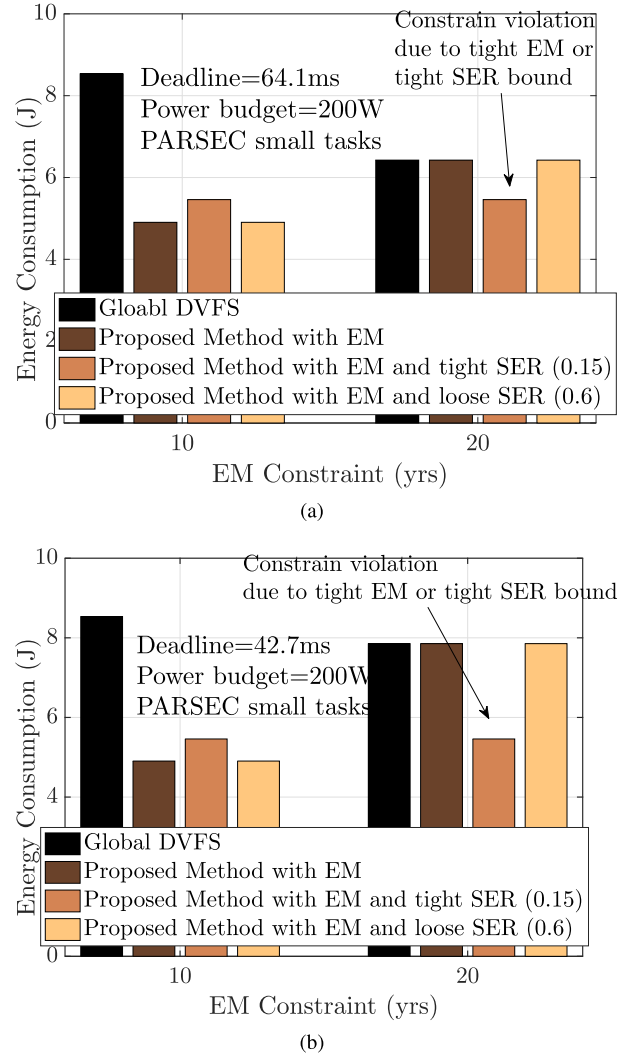


Fig. 9. Energy optimization results [Global DVFS, proposed with EM, and with/without tight and loose soft-error constraint from small task set on PARSEC benchmarks (different performance deadlines in (a) and (b))].

task set and a large task set cases are shown in Figs. 9 and 10, respectively. Each figure includes two groups with loose and tight EM, where the left four bars are loose EM constraints and the right four bars are tight EM constraints. In each group, the third bar is tight soft error constraints and the fourth bar is loose soft error constraints. As we can see, if we consider only EM constraints for the different EM lifetimes, power budgets, and performance deadlines for small task set case, the proposed method finds relatively high energy saving with both large and small performance deadlines [see Fig. 9(a) and (b)] than the global DVFS method because more cores can be in low power mode or turned OFF (dark silicon constraint) with the given power budgets and performance deadlines. When the loose soft error constraint is further considered ($SER_{sys} = 0.6$), our proposed method can still find good energy savings (similar or slightly higher than the method considering only EM constraint as shown in Figs. 9 and 10).

However, for the tight soft error constraint ($SER_{sys} = 0.15$) with tight EM constraint cases, our method violates the given EM constraints because soft errors and EM constraints have a

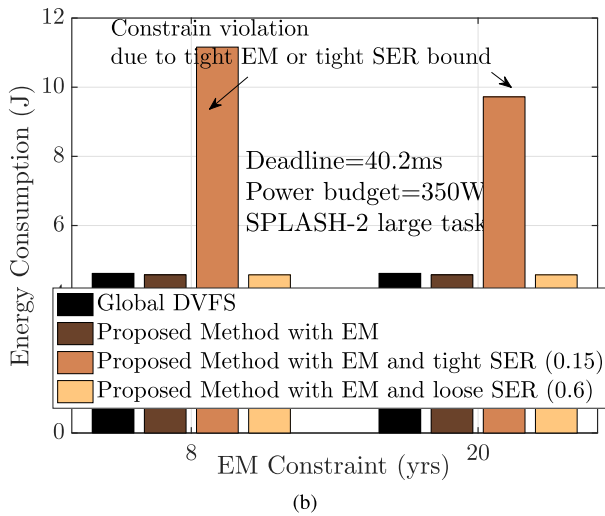
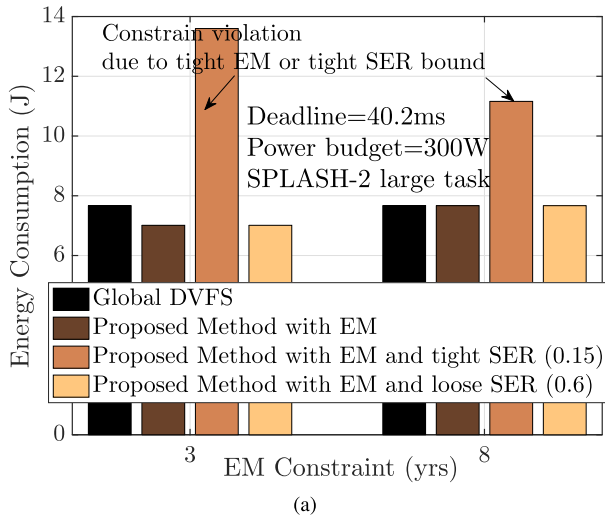


Fig. 10. Energy optimization results [Global DVFS, proposed with EM, and with/without tight and loose soft-error constraint from large task set on SPLASH-2 benchmarks (different power budgets in (a) and (b)).

contradictory relationship as seen in Fig. 7. For the large task set case in Fig. 10, more violations and less energy savings are observed. Those violations are caused by the higher task utilization of active cores, tight EM lifetime constraints, and power budget. However, our method can still find relatively good energy savings with the loose soft error constraints and tight EM constraints even at the large task sets. Thus, with exception to the violation cases (where both EM and soft error constraints are tight), our proposed method can find decent energy savings satisfying both EM and soft errors reliability constraints.

In Fig. 11, we further show the EM (a) and soft error (b) constrain violation cases under different EM and soft error constraints and power constraints (case 1–4 for small task sets, 5–8 for large task sets). In all cases 1–8, the soft error constraints are all tight. For cases 2, 4, and 8, the EM constraints are tight. First, as we can see, both EMs and soft errors are violated in cases 6 and 8 due to the tight soft error constraint with large task sets. For cases 2 and 4, only EMs are violated due to both tight EM and soft error

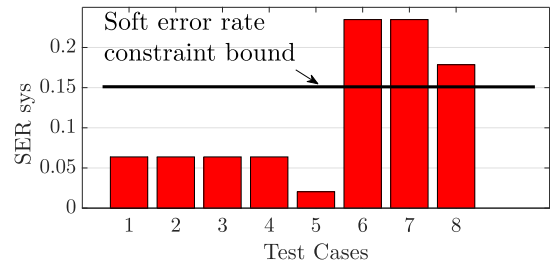
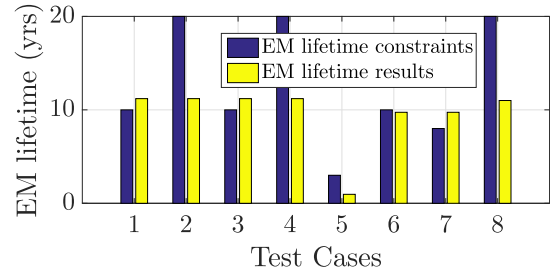


Fig. 11. Constraint violation cases.

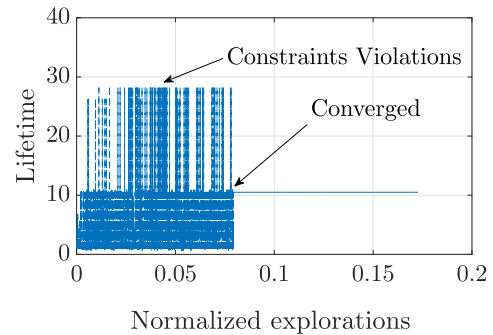


Fig. 12. Convergence rate of proposed DRM method with EM-induced lifetime constraint in 64-core Dark Silicon (SPLASH-2 Tasks).

constraints with small task sets. More interesting is in case 5 where EM is violated even the EM constraint is not tight. The reason is that we have very small power budget (tight power constraint). For case 7, which is large case, the soft error is very tight. As a result, soft error constraint is still violated even with large power budget. Therefore, as we can see, for energy-efficient computing on many-core systems with power constraints (dark silicon), and competing hard and soft error constraints, the results are no longer simple tradeoff among different factors, and instead, depend on those factors in a complicated way. In other words, under tightened power and performance constraints, we cannot satisfy both hard and soft errors at the same time and some other soft-error mitigation techniques become necessary in this case.

The proposed Q-learning method converges around 8% of explorations out of all possible state-action solution space as shown in Fig. 12. It also shows that system violation can be effectively prevented by our proposed penalty function (10).

To show the scalability of our proposed algorithm, we have added one more p-state on 64-core dark silicon chip with 64 tasks SPLASH-2 multithreaded benchmark. The full power

TABLE III
ELAPSED CPU TIME TO SOLVE THE PROPOSED Q-LEARNING
AND MILP PROBLEMS

64 tasks in 64-core dark silicon (total states)	Q-learning time (s)	MILP
Case 1 - 2 p-states (150 states)	0.172 s	9.1 s
Case 2 - 3 p-states (192 states)	0.183 s	88.08 s

TABLE IV
LARGE-SCALE EXPERIMENTS WITH FIVE P-STATE ON
128-CORE AND 256-CORE

# of core with 64 tasks with 5 p-states (total states)	Q-learning time (s)
Case 3 - 128-core (775 states)	0.699 s
Case 4 - 256-core (7160 states)	7.02 s

mode (2.0 GHz, 1.2 V) and the low power mode (1.0 GHz, 0.9 V), and very low power mode (800 MHz, 0.7 V) have been set for our framework. We compare two different numbers of p-states, case 1 has two p-states (full power and low power) and dark silicon states and case 2 has three p-states (full power, low power, very low power) and dark silicon states. Due to very low power mode, we choose relatively loose constraints (deadline = 12 ms, power budget = 350 W, EM = 5 years, SER = 0.6). Our proposed algorithm can find low energy consumption with lower p-state cases. We chose three p-states with the clustered DVFS [57], which have been employed to reduce the simulation time with small solution quality degradation due to a large number of cores in our experiment, so the case 1 is 150 states and the case 2 is 192 states. The case 2 has 28% more states, but the iteration requires only 6.58% more steps. Table III shows elapsed CPU time to solve the proposed Q-learning and MILP problem for each case.

Q-learning elapsed CPU time was obtained on iPython 5.1.0 with using only single core of Intel Xeon E5 system (clock 2.3 GHz) platform. Moreover, in order to show more higher scalability of the proposed algorithm, we increase p-states to five different ones for two large cases (128-core and 256-core), and their voltage and frequency (V , f) sets are (2.0 GHz, 1.2 V), (1.6 GHz, 1.1 V), (1.2 GHz, 1.0 V), (1.0 GHz, 0.9 V), and (800 MHz, 0.7 V) on 64 tasks SPLASH-2 multithreaded benchmark. We used 16-core as one cluster for the clustered DVFS. Also, we use 32-core as one cluster for dark silicon states in 128-core case and 64-core in 256-core case, which have been employed for a very large number of cores in our experiment. Table IV shows the elapsed CPU time results for solving the proposed Q-learning result and energy savings over global DVFS. 128-core with 5 p-states (case 3) has 775 different states, which is 516% more states than case 1, but the iteration requires only 406% more steps. 256-core with 5 p-states (case 4) has 7160 states, which is 4773% more states than case 1, but only 4081% more steps are required. Thus, our proposed DRM can be a scalable solution for a large number of cores and higher p-states cases since iteration steps and CPU times are close to increase linearly with respect to the total number of states.

Both large cases can also find decent energy savings. Due to extremely high cost of MILP method for large-case (128-core and 256-core with 5 p-states), the elapsed CPU time of Q-learning method has only been shown in Table IV.

V. CONCLUSION

In this paper, we proposed a new energy and lifetime optimization technique for emerging dark silicon manycore microprocessors considering hard and soft errors. The new approach was based on a newly proposed physics-based EM reliability model to predict the EM reliability of full-chip power grid networks for hard error. DVFS-aware SER model and the SOFR method are employed for system-level soft-error model, which has been widely used to estimate microprocessor level soft errors. We employed both DVFS and dark silicon core state using ON/OFF pulsing action as the two control knobs. The impact on DVFS for hard and soft errors was investigated. We focused on two optimization techniques for improving lifetime and reducing energy. To optimize lifetime, we first applied the adaptive Q-learning-based method, which was suitable for dynamic runtime operation as it was able to provide cost-effective yet good solutions. The second lifetime optimization approach was the MILP method, which typically yields better solutions but at higher computational costs. To optimize the energy of a dark silicon chip, we applied the Q-learning reinforcement learning method, which was suitable for our reliability management for the energy optimization considering hard and soft errors. Experimental results on a 64-core dark silicon chip showed that proposed methods work well for performance and lifetime optimizations considering the both soft and hard reliability constraints.

REFERENCES

- [1] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, E. Bassous, and A. R. LeBlanc, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE J. Solid-State Circuits*, vol. 9, no. 5, pp. 256–268, Oct. 1974.
- [2] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proc. 38th Annu. Int. Symp. Comput. Archit.*, New York, NY, USA, 2011, pp. 365–376.
- [3] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era," *Computer*, vol. 41, no. 7, pp. 33–38, Jul. 2008.
- [4] K. Chakraborty, "Over-provisioned multicore systems," Ph.D. dissertation, Dept. Comput. Sci., Univ. Wisconsin-Madison, Madison, WI, USA, 2008.
- [5] S. Cho and R. Melhem, "Corollaries to Amdahl's law for energy," *IEEE Comput. Archit. Lett.*, vol. 7, no. 1, pp. 25–28, Jan. 2008.
- [6] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Maestro: Orchestrating lifetime reliability in chip multiprocessors," in *Proc. 5th Int. Conf. High Perform. Embedded Archit. Compilers (HiPEAC)*, Berlin, Germany, 2010, pp. 186–200.
- [7] A. Das, R. A. Shafik, G. V. Merrett, B. M. Al-Hashimi, A. Kumar, and B. Veeravalli, "Reinforcement learning-based inter- and intra-application thermal optimization for lifetime improvement of multicore systems," in *Proc. 51st Annu. Design Autom. Conf.*, New York, NY, USA, 2014, pp. 170:1–170:6.
- [8] T. Kim, X. Huang, V. S. H.-B. Chen, and S. X.-D. Tan, "Learning-based dynamic reliability management for dark silicon processor considering EM effects," in *Proc. Design Autom. Test Eur. (DATE)*, Mar. 2016, pp. 463–468.
- [9] Y. Tan, W. Liu, and Q. Qiu, "Adaptive power management using reinforcement learning," in *Proc. Int. Conf. Comput.-Aided Design*, New York, NY, USA, 2009, pp. 461–467.

- [10] Y. Ge and Q. Qiu, "Dynamic thermal management for multimedia applications using machine learning," in *Proc. 48th Design Autom. Conf.*, New York, NY, USA, 2011, pp. 95–100.
- [11] H. Shen, J. Lu, and Q. Qiu, "Learning based DVFS for simultaneous temperature, performance and energy management," in *Proc. 13th Int. Symp. Quality Electron. Design (ISQED)*, Mar. 2012, pp. 747–754.
- [12] R. Ye and Q. Xu, "Learning-based power management for multi-core processors via idle period manipulation," in *Proc. 17th Asia South Pacific Design Autom. Conf. (ASP-DAC)*, Jan. 2012, pp. 115–120.
- [13] [2014]. *International Technology Roadmap for Semiconductors (ITRS)*, [Online]. Available: <http://public.itrs.net>
- [14] C. Prasad *et al.*, "Self-heat reliability considerations on Intel's 22 nm tri-gate technology," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2013, pp. 5D.1.1–5D.1.5.
- [15] H. Amrouch, V. M. van Santen, T. Ebi, V. Wenzel, and J. Henkel, "Towards interdependencies of aging mechanisms," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2014, pp. 478–485.
- [16] J. R. Black, "Electromigration—A brief survey and some recent results," *IEEE Trans. Electron Devices*, vol. 16, no. 4, pp. 338–347, Apr. 1969.
- [17] I. A. Blech, "Electromigration in thin aluminum films on titanium nitride," *J. Appl. Phys.*, vol. 47, no. 4, pp. 1203–1208, Apr. 1976.
- [18] X. Huang, T. Yu, V. Sukharev, and S. X.-D. Tan, "Physics-based electromigration assessment for power grid networks," in *Proc. Design Autom. Conf. (DAC)*, Jun. 2014, pp. 1–6.
- [19] V. Sukharev, "Beyond Black's equation: Full-chip EM/SM assessment in 3D IC stack," *Microelectron. Eng.*, vol. 120, pp. 99–105, Sep. 2014.
- [20] V. Sukharev, A. Kteyan, E. Zschech, and W. D. Nix, "Microstructure effect on EM-induced degradations in dual inlaid copper interconnects," *IEEE Trans. Device Mater. Rel.*, vol. 9, no. 1, pp. 87–97, Mar. 2009.
- [21] Z. Suo, *Reliability of Interconnect Structures of Comprehensive Structural Integrity*, vol. 8. Amsterdam, The Netherlands: Elsevier, 2003.
- [22] S. Chatterjee, M. Fawaz, and N. F. Najm, "Redundancy-aware electromigration checking for mesh power grids," in *Proc. Int. Conf. Comput. Aided Design (ICCAD)*, 2013, pp. 540–547.
- [23] X. Huang, A. Kteyan, X. Tan, and V. Sukharev, "Physics-based electromigration models and full-chip assessment for power grid networks," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 35, no. 11, pp. 1848–1861, Feb. 2016.
- [24] Z. Lu, W. Huang, J. Lach, M. Stan, and K. Skadron, "Interconnect lifetime prediction under dynamic stress for reliability-aware design," in *Proc. Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2004, pp. 327–334.
- [25] A. Das, A. Kumar, and B. Veeravalli, "Reliability-driven task mapping for lifetime extension of networks-on-chip based multiprocessor systems," in *Proc. Conf. Design, Autom. Test Eur.*, San Jose, CA, USA, 2013, pp. 689–694.
- [26] *Failure Mechanisms and Models for Semiconductor Devices, JEDEC Publication JEP122C*, accessed on Sep. 2016. [Online]. Available: <http://www.jedec.org>
- [27] D. Zhu, R. Melhem, and D. Mosse, "The effects of energy management on reliability in real-time embedded systems," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, Washington, DC, USA, Sep. 2004, pp. 35–40.
- [28] L. Tan, S. Song, P. Wu, Z. Chen, R. Ge, and D. Kerbyson, "Investigating the interplay between energy efficiency and resilience in high performance computing," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. (IPDPS)*, May 2015, pp. 786–796.
- [29] Y. Zhang, K. Chakrabarty, and V. Swaminathan, "Energy-aware fault tolerance in fixed-priority real-time embedded systems," in *Proc. Int. Conf. Comput. Aided Design, (ICCAD)*, Nov. 2003, pp. 209–213.
- [30] J. Srinivasan, S. V. Adve, P. Bose, J. Rivers, and C.-K. Hu, *Ramp: A Model for Reliability Aware Microprocessor Design*. Poughkeepsie, NY, USA: IBM, 2003.
- [31] X. Li, S. Adve, P. Bose, and J. Rivers, "Architecture-level soft error analysis: Examining the limits of common assumptions," in *Proc. 37th Annu. IEEE/IFIP Int. Conf. Dependable Syst. Netw.*, Jun. 2007, pp. 266–275.
- [32] S. S. Mukherjee, C. Weaver, J. Emer, S. K. Reinhardt, and T. Austin, "A systematic methodology to compute the architectural vulnerability factors for a high-performance microprocessor," in *Proc. 36th Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO)*, Washington, DC, USA, 2003, p. 29.
- [33] A. Dixit and A. Wood, "The impact of new technology on soft error rates," in *Proc. IEEE Int. Rel. Phys. Symp. (IRPS)*, Apr. 2011, pp. 5B.4.1–5B.4.7.
- [34] E. Ibe, H. Taniguchi, Y. Yahagi, K. I. Shimbo, and T. Toba, "Impact of scaling on neutron-induced soft error in SRAMs from a 250 nm to a 22 nm design rule," *IEEE Trans. Electron Devices*, vol. 57, no. 7, pp. 1527–1538, Jul. 2010.
- [35] Y. Cao, *Predictive Technology Model for Robust Nanoelectronic Design*. New York, NY, USA: Springer-Verlag, 2011.
- [36] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
- [37] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.
- [38] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Comput.*, vol. 6, pp. 1185–1201, Nov. 1994.
- [39] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2011, pp. 52:1–52:12.
- [40] N. Binkert *et al.*, "The gem5 simulator," *ACM SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, 2011.
- [41] M. M. K. Martin *et al.*, "Multifacet's general execution-driven multiprocessor simulator (GEMS) toolset," *ACM SIGARCH Comput. Archit. News Lett.*, vol. 33, no. 4, pp. 92–99, 2005.
- [42] A. Patel, F. Afram, S. Chen and K. Ghose, "MARSS: A full system simulator for multicore x86 CPUs," in *Proc. 48th ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, New York, NY, USA, 2011, pp. 1050–1055.
- [43] T. F. Wenisch, R. E. Wunderlich, M. Ferdman, A. Ailamaki, B. Falsafi, and J. C. Hoe, "SimFlex: Statistical sampling of computer system simulation," *IEEE Micro*, vol. 26, no. 4, pp. 18–31, Jul. 2006.
- [44] J. H. Ahn, S. Li, O. Seongil, and N. P. Jouppi, "McSimA+: A manycore simulator with application-level+ simulation and detailed microarchitecture modeling," in *Proc. IEEE Int. Symp. Perform. Anal. Syst. Softw. (ISPASS)*, Apr. 2013, pp. 74–85.
- [45] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proc. 17th Int. Conf. Parallel Archit. Compilation Techn.*, New York, NY, USA, 2008, pp. 72–81.
- [46] S. Woo, M. O. Ohara, E. Torrie, J. Singh, and A. Gupta, "The splash-2 programs: Characterization and methodological considerations," in *Proc. 22nd Annu. Int. Symp. Comput. Archit.*, Jun. 1995, pp. 24–36.
- [47] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *Proc. Int. Symp. Comput. Archit.*, 2003, pp. 2–13.
- [48] M. L. Littman, T. L. Dean, and L. P. Kaelbling, "On the complexity of solving Markov decision problems," in *Proc. 11th Conf. Uncertain. Artif. Intell.*, San Francisco, CA, USA, 1995, pp. 394–402.
- [49] K. He, X. Huang, and S. X.-D. Tan, "EM-based on-chip aging sensor for detection and prevention of counterfeit and recycled ICs," in *Proc. Int. Conf. Comput. Aided Design (ICCAD)*, Nov. 2015, pp. 1–6.
- [50] M. Berkold and T. Tian. (2010). *Cpu Monitoring With DTS/PECI*. [Online]. Available: <http://www.intel.com/content/www/us/en/embedded/testing-and-validation/cpu-monitoring-dts-peci-paper.html>
- [51] W. Wu, L. Jin, J. Yang, P. Liu, and S. X.-D. Tan, "A systematic method for functional unit power estimation in microprocessors," in *Proc. Design Autom. Conf. (DAC)*, Jun. 2006, pp. 554–557.
- [52] G. Dhiman and T. S. Rosing, "Dynamic power management using machine learning," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design*, New York, NY, USA, 2006, pp. 747–754.
- [53] H. Jung and M. Pedram, "Supervised learning based power management for multicore processors," *IEEE Trans. Comput.-Aided Design Integr.*, vol. 29, no. 9, pp. 1395–1408, Sep. 2010.
- [54] Z. Chen and D. Marculescu, "Distributed reinforcement learning for power limited many-core system performance optimization," in *Proc. Design, Autom. Test Eur. Conf. Exhibit.*, San Jose, CA, USA, 2015, pp. 1521–1526.
- [55] M. Sridharan and G. Tesauro, *Multi-Agent Q-Learning and Regression Trees for Automated Pricing Decisions*. Boston, MA, USA: Springer, 2002, pp. 217–234.
- [56] G. Tesauro and J. O. Kephart, "Pricing in agent economies using multi-agent Q-learning," *Auto. Agents Multi-Agent Syst.*, vol. 5, no. 3, pp. 289–304, 2002.
- [57] T. Kolpe, A. Zhai, and S. Sapatnekar, "Enabling improved power management in multicore processors through clustered DVFS," in *Proc. Design, Autom. Test Eur. (DATE)*, Mar. 2011, pp. 1–6.
- [58] Hewlett-Packard, Intel, Microsoft, Phoenix, and Toshiba. (2013). *Advanced Configuration and Power Interface Specification 5.0a*. [Online]. Available: <http://www.acpi.info>



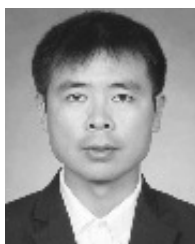
Taeyoung Kim (S'10) received the M.S. degree in electrical and computer engineering from the University of Virginia, Charlottesville, VA, USA, in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of California, Riverside, CA, USA.

His current research interests include embedded system design, electronic design automation, reliability-aware system modeling, simulation, and optimization.



Zeyu Sun (S'16) received the B.S. degree in electronic and computer engineering with the Hong Kong University of Science and Technology, Hong Kong, in 2015. He is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering, University of California, Riverside, CA, USA.

His current research interests include electromigration modeling and assessment and reliability-aware performance optimization.

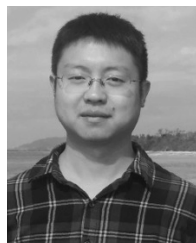


Hai-Bao Chen (M'16) received the B.S. degree in information and computing sciences, and the M.S. and Ph.D. degrees in applied mathematics from Xi'an Jiaotong University, Xi'an, China, in 2006, 2008, and 2012, respectively.

He joined Huawei Technologies, Shenzhen, China, where he focused on cloud computing and big data. He was a Post-Doctoral Research Fellow with the Electrical Engineering Department, University of California, Riverside, CA, USA, from 2013 to 2014. He is currently an Assistant Professor with

the Department of Micro/Nano-electronics, Shanghai Jiao Tong University, Shanghai, China. He has authored or co-authored more than 20 papers in scientific journals and conference proceedings. His current research interests include model order reduction, system and control theory, circuit simulation, cloud computing and big data, and electromigration reliability.

Dr. Chen was a recipient of the Best Paper Award nomination from Asia and South Pacific Design Automation Conference in 2015.

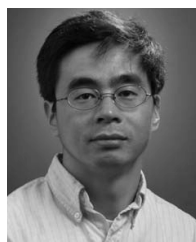


Hai Wang received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2007, and the M.S. and Ph.D. degrees from the University of California, Riverside, CA, USA, in 2008 and 2012, respectively.

He has been an Associate Professor with the University of Electronic Science and Technology of China, Chengdu, China. He has authored over 30 peer-reviewed international conference papers and journal articles in related research field. His current research interests include electrical/thermal

verification and optimization of very large scale integration circuits and systems.

Dr. Wang serves as a Technical Program Committee Member of several international conferences including DATE, ASP-DAC, and ISQED, and also serves as a reviewer for many journals including IEEE TRANSACTIONS ON COMPUTERS, IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, and ACM TRANSACTIONS ON DESIGN AUTOMATION OF ELECTRONIC SYSTEMS.



Sheldon X.-D. Tan (S'96–M'99–SM'06) received the B.S. and M.S. degrees in electrical engineering from Fudan University, Shanghai, China, in 1992 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Iowa, Iowa City, IA, USA, in 1999.

He is currently an Associate Director of Computer Engineering Program and a Professor with the Department of Electrical Engineering, University of California, Riverside, CA, USA. His current research interests include very large scale integration reliability modeling, optimization and management at circuit and system levels,

thermal modeling, optimization and dynamic thermal management for many-core processors, statistical modeling, simulation and optimization of mixed-signal/RF/analog circuits, and parallel circuit simulation techniques based on graphical processing unit and multicore systems.

Dr. Tan was a recipient of the NSF CAREER Award in 2004, the Outstanding Oversea Investigator Award from the National Natural Science Foundation of China in 2008, the Best Paper Award from the 2007 IEEE International Conference on Computer Design (ICCD'07), the Best Paper Award from the 1999 IEEE/ACM Design Automation Conference, three Best Paper Award Nomination from the IEEE/ACM Design Automation Conferences in 2005, 2009, and 2014, and the Best Paper Award nomination from the ASP-DAC in 2015. He is currently the Editor-In-Chief for Integration of the *The VLSI Journal*. He is also an Associate Editor of three journals: the IEEE TRANSACTIONS ON VLSI SYSTEMS and the *ACM Transactions on Design Automation of Electronic Systems*.