Learning-Based Dynamic Reliability Management For Dark Silicon Processor Considering EM Effects

Taeyoung Kim*, Xin Huang†, Hai-Bao Chen§, Valeriy Sukharev‡, Sheldon X.-D. Tan†
* Department of Computer Science and Engineering, University of California, Riverside, CA 92521, USA
† Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA
‡ Mentor Graphics Corporation, Fremont, CA 94538, USA
§ Department of Micro/Nano-electronics, Shanghai Jiao Tong University, Shanghai 200240, China

***Abstract*—In this article, we propose a new dynamic reliability management (DRM) technique for emerging dark silicon many-core processors. We formulate our DRM problem as minimizing the energy consumption subject to the reliability, performance and thermal constraints. The new approach is based on a newly proposed physics-based electromigration (EM) reliability model to predict the EM reliability of full-chip power grid networks. We consider thermal design power (TDP) as the power constraint for a dark silicon manycore processor. We employ both dynamic voltage and frequency scaling (DVFS) and dark silicon core using ON/OFF pulsing action as the two control knobs. To solve the problem, we apply the adaptive Q-learning based method, which is suitable for runtime operation as it can provide cost-effective yet good solutions. A large class of multithreaded applications is used as the benchmark 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 reduce the energy consumption of a dark silicon manycore system when the system is not tightly constrained. The proposed method can outperform a simple global DVFS method significantly in this case.**

I. Introduction

Technology scaling has led to the continuous integration of devices, and future manycore processors will have more cores integrated. However due to the diminishment of Dennard's scaling [1], the power density of chips starts to increase for current and future technology nodes. The consequence is the emerging of so-called dark silicon manycore processors as only a percentage of cores can be powered on the chip due to the power and temperature limitations. Recently, architecture researchers have begun focusing on the development of manycore processors with as many as 100 and 1000-core dark silicon manycore processors on a single die. Such manycore systems pose new challenges and opportunities for power/thermal and reliability management of those chips [2].

Dark silicon needs to perform under the lowest possible energy consumption as it is limited by the energy available. Power, performance and temperature limitations are traditional dominant factors in the chip for energy efficient high performance computing. Recently, reliability is becoming a limiting constraint in high-performance nanometer VLSI chip designs due to the high failure rates in deep submicron and nanoscale devices. It is expected that future chips will show signs of reliability-induced aging much sooner than the previous generations. Among many reliability effects, electromigration (EM)-induced reliability has become a major design constraint due to aggressive transistor scaling and increasing power density. For dark silicon, the reliability can become worse as cores will experience more thermal cycles during the on-off operations. A manycore processor also may operate in the very low voltage or even near threshold voltage regions which also hurts the soft-error induced reliability. For the EM effects, however, dark silicon can provide one more knob (turn on or off for a core) to increase energy saving with considering the EM-induced lifetime of the chip, which will be explored in this work.

Existing studies mainly focus on manycore or dark silicon architecture such as core organization, topology, optimal number of cores, and workload management, such as task allocation, migration, and scheduling [2]–[5]. Most of those existing works consider on power, temperature, and performance for energy efficiency and low power system. Recently, reliability-aware management with dark silicon effect on manycore scaling have been proposed [6]–[10]. However, all of these works considered general reliability models, which will not be accurate for specific failure mechanism.

In this work, we propose a new dynamic reliability management (DRM) technique for emerging dark silicon manycore processors. We formulate our DRM problem as minimizing the energy consumption subject to the reliability, performance and thermal constraints. We focus on the electromigration-induced reliability problem as it is the dominant failure effect for on-chip interconnects and the proposed techniques are orthogonal to other reliability effects. The new approach is based on a newly proposed physics-based electromigration (EM) reliability model to predict the EM reliability of full-chip power grid networks.

We consider thermal design power (TDP) as the power constraint for a dark silicon manycore processor. We employ both dynamic voltage and frequency scaling (DVFS) and dark silicon core using ON/OFF pulsing action as the two control knobs. To solve the proposed energy optimization problem, we apply the adaptive Q-learning based method, which is suitable for runtime operation as it can provide cost-effective yet good solutions. Our implementation framework also consists of interval core-based microarchitecture model [11], and X86-based power model. We also apply HotSpot thermal model for the temperature estimation [12].

A large class of multithreaded applications is used as the benchmark to validate and compare the proposed dynamic reliability management methods. Experimental results on a 64-core dark silicon manycore processor show that the proposed DRM algorithm can effectively reduce the energy consumption of dark silicons under the given less constrained lifetime, power budget and performance constraints. The proposed method can outperform a simple global DVFS method significantly in this case as well.

This work is supported in part by NSF grant under No. CCF-1527324, in part by NSF Grant under No. CCF-1255899, in part by Semiconductor Research Corporation (SRC) grant under No. 2013-TJ-2417.

978-3-9815370-7-9/DATE16/ ©2016 EDAA

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 the 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.

Currently EM effects are mainly modeled by empirical Black's equation [13] and Blech limit [14]. The primary drawbacks of those models are that they are not physics-based, which means that they lack predictability for varying stressed conditions and for complicated wire structures. These models also do not consider the inherent redundancy in power grid networks, which are the most vulnerable interconnects in a chip.

To mitigate those problems, a more physics-based compact EM model has been proposed recently for full-chip reliability analysis [15], [16], 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}) 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 the evolution of vacancy and plated atom concentrations (see, for example [17]) in the confined 1D line

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

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 power/ground (p/g) network becomes a time-varying network, and its voltage drops will keep changing over the time [15].

A. 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 [18]. 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 nodal size. In our problem, the time scale is the EM time scale, which can be months or years.

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

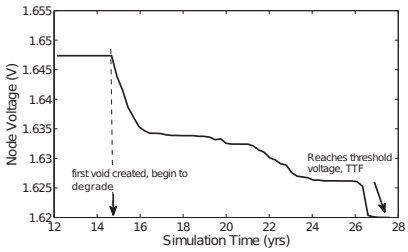


Fig. 1. Voltage of the first failed node in different simulation time

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 exceed the critical voltage drops allowed (say 10% of Vdd). For our dark silicon manycore systems, we use the same mesh-structured p/g network for all the cores.

B. System level EM-reliability model

At the system level, the manycore 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 a constant temperature. The previous study shows that whole system MTTF, $MTTF_{sys}$, (expected lifetime) under different temperature can be approximated by [19]:

$$MTTF_{sys} = \frac{1}{(\sum_{k=1}^{n}(\Delta t_k \frac{1}{MTTF_{R,k}}))/T} \quad (2)$$

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. To consider a system-level reliability on a manycore dark silicon processor, we use the shortest lifetime among all the cores as the lifetime for all manycore processors [20], [21].

III. New dynamic reliability management method for dark silicon

In this section, we formulate our new dynamic reliability management (DRM) problem as minimizing energy considering a EM-induced lifetime of dark silicon manycore processors by controlling the number of active cores and the suitable performance state (p-state) subject to power budget, performance deadline, and temperature constraints.

A. Q-learning based formulation and solution

1) State and action determination: Q-learning [22], a reinforcement learning method, performs the control by maximizing expected long-term rewards [23]. Q-learning can handle problems with stochastic transition and it has been proved that this method is able to converge close to the optimal approximation of state-action function for arbitrary policy [24]. In our problem, the state (s) consists of the configurations of DVFS and active core status (on/off) for each core. DVFS uses performance state (p-state) which can represent operating voltage and frequency. Action (a) is defined as a state transition from one state to the another state. Transiting an action in a state makes the agent with a reward (negative penalty) scoring that is calculated with the quantity of state-action combination (Q). Q can be defined as a set of states (S) and a set of action (A) table, $S \times A$, which is Q-table. Q-table can be updated by a Q-value function which a long-term penalty function with state and action.

Fig. 2 shows proposed Q-learning based reliability optimization framework. The environment part is dark silicon manycore processor, another is learning agent, which is Q-learning algorithm. The learning agent can obtain the environmental state, calculate penalty function, and finally, decide the next action.

Table I illustrates an example of state, p-state, and active core for small 3-core dark silicon chip. In p-state, 1 is low power mode, 2 is full power mode, and 0 means the core is turned off. Clearly, state 0 is the state with a minimum number of active cores, which are in the lowest power modes and state 8 is the state with a maximum number of active core, which are in the highest power modes.

TABLE I
AN EXAMPLE OF CONTROL STATES FOR A 3-CORE PROCESSOR

State	p-state	active core	State	p-state	active core
0	0,0,1	off,off,on	1	0,0,2	off,off,on
2	0,1,1	off,on,on	3	0,1,2	off,on,on
4	0,2,2	off,on,on	5	1,1,1	on,on,on
6	1,1,2	on,on,on	7	1,2,2	on,on,on
8	2,2,2	on,on,on			

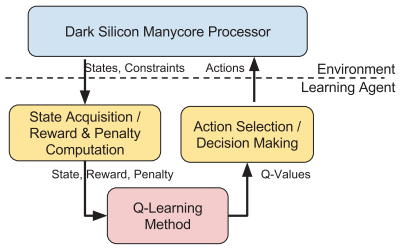


Fig. 2. Q-Learning model with reliability-aware dark silicon framework

2) Q-value function and Q-learning process: In the Q-learning process, one critical issue is to define the Q-value function with penalty term. Specifically, let's formally define *State* i: $s_i = \{PS_i, CS_i)$. PS_i is the set of p-state (DVFS setting) for all cores. CS_i is the set of core status for all cores. Each state s_i will determine the total power of the whole chip $Power(s_i)$, worse case performances of all the cores $Perf_{max}(s_i)$, the maximum temperature incurred $Temp_{max}(s_i)$, the minimum lifetime among cores, $EM_{min}(s_i)$ defined as the lifetime of the chip, and $E_(s_i)$ is total core energy consumption in the whole chip. Total core energy consumption can be obtained from $\sum_k E_k(s_i)$ which is k-th core's core energy, each core's energy can be calculated by $Power_k(s_i) \times Perf_k(s_i)$ where $Power_k(s_i)$ is average k-th core's power and $Perf_k(s_i)$ is each k-th core's performance. An action, say $a_{i,j}$, can be viewed as the transition from state i to state j. Then the penalty function Q determines a penalty and a new state which is related with the previous state and selected action. New updated Q-value at every step Δt can be expressed in an iterative way [23]:

$$Q^{t+1}(s(t), a(t)) = Q^t(s(t), a(t)) + \alpha(t) \times \left(PT(t+1) + \gamma \min_a (\forall Q^t(s(t+1), a)) - Q^t(s(t), a(t)) \right) \quad (3)$$

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 [23]. $s(t+1)$ is determined by action $a(t)$, so $Q^t(s(t+1), a)$ are all possible action's 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 ($PT(t+1) + \gamma \min_a(\forall Q^t(s(t+1), a))$) updates the new Q-value (Q^{t+1}) with the learning rate.

The penalty term, ($PT(t+1)$) in (3) 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 four main constraints: EM-induced lifetime, total core power, performance deadline of all the tasks, and upper temperature limit. Total core energy consumption is objective that we want to maximize. As a result, we define the penalty function PT in [25] to consider multiple constraints.

We can build a penalty term (PT) as shown in (4). PT_E is a penalty term for total core energy, PT_{EM} is a penalty term for EM-induced lifetime, PT_{power} for power, PT_{temp} for temperature, and PT_{perf} for performance deadline of all tasks. Each penalty term (PT_x) is normalized in (4). We use feature scaling method to bring all values between 0 and 1. For instance $PT_E = \frac{E(t+1) - E(t)}{E_{Max} - E_{Min}}$ 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, $PT_{EM} = \frac{MTTF(t) - MTTF(t+1)}{MTTF_{Max} - MTTF_{Min}}$ for EM related penalty, where $MTTF(t)$ is the MTTF of the system in the previous time t and $MTTF(t+1)$ is the MTTF of the system at current $t+1$.

$$PT = PT_E + C \sum_{x=\{EM, power, temp, perf\}} \delta_x PT_x$$
$$\delta_x = \begin{cases} 0 & \text{if } PT_x \leq B_x + \Delta_x \\ 1 & \text{if } PT_x > B_x + \Delta_x \end{cases} \quad (4)$$

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

respectively. Each Δ_x is the difference between each bound and average penalty (PT) for power, performance, and temperature. Δ_x is positive if the system violates the given constraint, otherwise, it is negative, and the system is bounded and performs well. Therefore, if the system has violated the user constraints in the past, then the penalty can be more highly weighted (due to large value for constant C in (4)). With the given state, action, penalty function, we can update the Q-table as explained in the Algorithm 1.

Algorithm 1 Learning-based dynamic reliability management (DRM)

Input: A initial state set for each core (p-state and core status).
Output: The selected p-state and core status (on/off) for each core.
1: Initialize all Q-values in the Q-table to zero.
2: Denote the current state as $s(t)$, Find an action $a(t)$ with the lowest Q^t, and switch to next state with the corresponding p-state and active core.
3: Evaluate and update environment, such as energy, lifetime, performance, temperature, and power. Then, calculate the corresponding new penalty $PT(t+1)$ and then update Q^{t+1}.
4: Set the current state as a new action and iterate from Step 2.
5: When all Q-values changes are less than the certain threshold, then this is considered as the optimal policy chosen.

B. 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. 3. We first describe the major component models of the framework such as microarchitecture, power estimation, thermal and reliability models. Our proposed framework uses Sniper as a microarchitecture model, which is an accurate and fast application-level interval-based microarchitecture simulation [11]. 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) [26], GEMS [27], MARSSx86 [28] and SimFlex [29] 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 [30]. Thus, to support dark silicon and manycore processor, we choose application-level Sniper simulator. This Sniper interval-based model is accurately matching well with the Intel x86 multi-core architecture [11]. PARSEC [31] and SPLASH-2 [32] benchmarks are used for our platform workloads. We use both workloads to evaluate our proposed framework and algorithm in Section IV.

For the power/energy 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 multi-threaded and multi-core processor models. At each step of performance measurement in Sniper, McPAT can estimate the power and energy consumption. For the thermal model, we use HotSpot to accurately characterize the thermal traces from the given multithreaded task run in each core [33]. To enable the dark silicon feature, the floor plan, and power trace

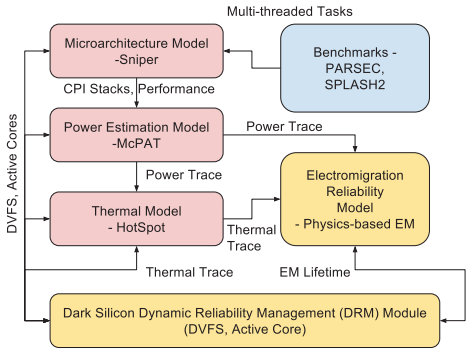
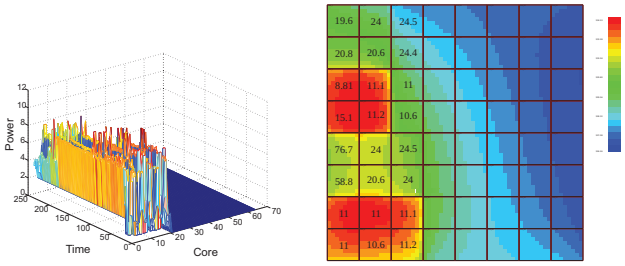


Fig. 3. The evaluation platform for dark silicon and DRM algorithms

are dynamically controlled by the dark silicon DRM module in Fig. 3

As shown in Fig. 3, once the cycle per instruction (CPI) stacks and power/energy traces are achieved in the microarchitecture model with the power model, the thermal model can generate thermal traces for given task run. With each core's power trace, thermal trace, core voltage, core frequency, and active cores, we can perform EM reliability analysis and the system-level assessment for processor lifetime based on the reliability models. Fig. 4(a) and 4(b) show the results from the proposed framework, which are the power traces, thermal measurement, and EM lifetime on a 64-core dark silicon chip. There are 20-core-enabled at the normal DVFS setting (2.0Ghz, 1.2V) and 64 multi-threaded tasks (16x CHOLESKYs, 16x RADIXs, 16x RAYTRACEs, 16x VOLRENDs) on a 64-core dark silicon chip. Fig. 3 only shows the core area. Power budgeting is not applied here.



(a) (b)

Fig. 4. (a) SPLASH2 benchmark 64 multithreaded tasks power traces with 44 cores off(b) Thermal (color:degree) and EM lifetime (number:yrs) analysis on 64 cores

IV. Numerical Results and Discussions

A. Evaluation setup

The proposed new DRM algorithms and evaluation platform for dark silicon has been implemented in Python 2.7.9 with numerical libraries (Numpy 1.9.2 and Scipy 0.15.1). We revised the architecture simulator (Sniper 6.1), power model simulator (McPAT 1.0.32), and thermal simulator (HotSpot 5.02 [33]) to estimate reliability-aware performance and lifetime task models on top of new physics-based EM model [15] for manycore processors and added the ability to dynamically turn-off partial cores. In the evaluation platform shown in Fig 3, each simulator module is connected with each custom

plugin connector (Python 2.7.9) so that one simulator's result can dynamically feed the other's inputs. DRM module controls each simulation model, DVFS, and active-core status. In Fig 2, the learning agent and Q-learning method have been implemented in Python 2.7.9 with extensive use of Numpy for matrix operations.

Our framework has been validated with a 64-core processor model on the Parsec and SPLASH-2 multi-threaded application benchmarks. We use two task set cases, a small case is a small number of tasks with PARSEC benchmark (1 BLACKSCHOLES, 1 CANNEAL, 1 FREQMINE, and 1 VIPS). For a large case, a large number of tasks with SPLASH2 benchmark (16 CHOLESKYs, 16 RADIXs, 16 RAYTRACEs, 16 VOLRENDs) are used. Each case has the same 64 threads.

In this experiment, we choose two performance states for DVFS, one is full power mode (2.0GHz, 1.2V setting) and another is low power mode (1.0Ghz, 0.9V setting) for our framework, which is controlled by dark silicon DRM module in Fig 3. We follow ACPI standard and Enhanced Intel Speedstep Technology [34] with 45nm technology.

Due to the large number of cores (64-core processors) with two DVFS states, we group 4 cores as one cluster and the cores in one cluster have the same p-state and core status (Clustered DVFS [35]). In this way, we can reduce the simulation time with small solution quality degradation. Every time, we can turn on or off four cores at a time, so there are 150 possible states for 64-core dark silicon chip in our experiment. For the multi-tasks, we use pinned scheduler, which is the interleaving of round-robin models implemented in Sniper Sim [11]. To show that our DRM can find lowest possible energy consumption for the given constraints, we compare our results with global (Per-chip) DVFS method, which has the smallest overhead and largest control granularity, for dark silicon platforms. In this case, all active cores will have the same p-state (if they are active cores) under lifetime, power budget, and performance deadline.

B. Evaluation of the proposed Q-Learning DRM method

First, we evaluate our learning-based DRM method (see Section III) by energy saving with different sets of EM lifetime constraints, power budgets and performance deadline. Fig. 5 and Fig. 6 show the energy saving given EM-induced lifetime constraint, power budget and performance deadline for small and large task sets on a 64-core dark silicon chip. As we can see energy saving for the different lifetime and performance constraints in Fig. 5, our method finds relatively high energy savings (37.5% and 18.1%) with large performance deadline (64.1ms) than the global DVFS method and core status because the 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 (42.7ms), there is still a chance to highly save energy (37.5%) than global DVFS in the smaller lifetime constraint (10 yr). However, for the higher lifetime and smaller performance constraints, energy saving will be close to global DVFS method as shown in Fig. 5. This can be explained that there are fewer rooms left for saving energy due to the tight constraints.

For the large task set case, energy saving will be limited as shown in Fig. 6 and be close to simple DVFS result even if there is still energy saving (8.5%) with lower EM lifetime constraint. With higher power budget, more energy can be saved (40.3%) because performance can be increased with more cores turned on. This indicates significant energy saving can be made for both small and large tasks with given lifetime, power budget and performance deadline.

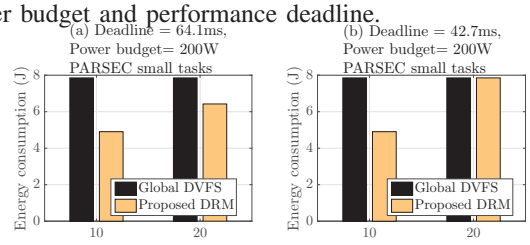


Fig. 5. Energy optimization with global DVFS (all cores are in the same p-state) and our proposed DRM on PARSEC small task set - different performance deadline and EM lifetime constraints)

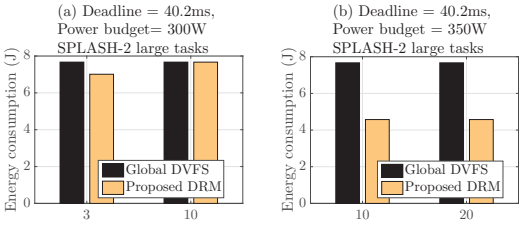


Fig. 6. Energy optimization with global DVFS (all cores are in the same p-state) and our proposed DRM on SPLASH-2 tasks - different power budget and EM lifetime constraints)

Fig. 7 shows the lifetime, power consumptions and performances from our proposed DRM method and it indicates all the results can meet the given lifetimes, power budgets and performance deadlines. Furthermore, no violations were found in either small (test case 1–4) or large task (test case 5–8) set results in Fig. 7.

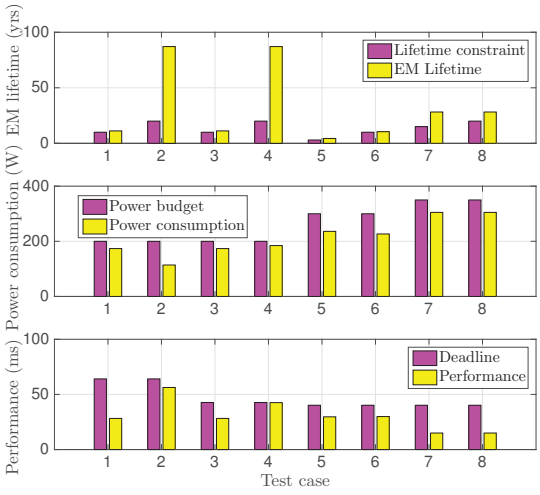


Fig. 7. Q-learning constraints from test case (1–4) PARSEC tasks and light (5,8) SPLASH-2 tasks in 64-core dark silicon chip

The proposed Q-learning method converges around 8% of explorations out of all possible state-action solution space as shown in Fig. 8. It also shows that system violation can be effectively prevented by our proposed penalty function (4).

V. Conclusion

In this article, we have proposed a new dynamic reliability management (DRM) technique for emerging dark silicon

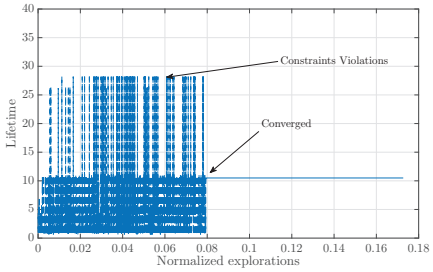


Fig. 8. Convergence rate of proposed DRM method with EM-induced lifetime constraint in 64-core Dark Silicon (SPLASH-2 Tasks)

manycore processors. We formulated our DRM problem as minimizing the energy consumption subject to the reliability, performance and thermal constraints. The new approach is based on a newly proposed physics-based electromigration (EM) reliability model to predict the EM reliability of full-chip power grid networks. We have employed both dynamic voltage and frequency scaling (DVFS) and dark silicon core using ON/OFF pulsing action as the two control knobs. To solve the problem, we applied the adaptive Q-learning based method, which is suitable for runtime operation as it can provide cost-effective yet good solutions. Experimental results on a 64-core dark silicon chip show that the proposed DRM algorithm can effectively reduce the energy consumption of a dark silicon chip under the given lifetime constraint, power budget and performance limit. When dark silicon manycore systems are not tightly constrained, the proposed method can outperform a simple global DVFS method significantly.

REFERENCES

[1] R. Dennard, F. Gaensslen, H. Yu, V. Rideout, E. Bassous, and A. LeBlanc, "Design of ion-implanted mosfet's with very small physical dimensions," *IEEE Journal of Solid-State Circuits*, vol. 9, pp. 256–268, October 1974.

[2] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *Proceedings of the 38th Annual International Symposium on Computer Architecture*, ISCA '11, (New York, NY, USA), pp. 365–376, ACM, 2011.

[3] K. Chakraborty, *Over-provisioned Multicore Systems*. PhD thesis, Madison, WI, USA, 2008. AAI3327881.

[4] S. Cho and R. Melhem, "Corollaries to amdahl's law for energy," *IEEE Comput. Archit. Lett.*, vol. 7, pp. 25–28, Jan. 2008.

[5] M. D. Hill and M. R. Marty, "Amdahl's law in the multicore era," *Computer*, vol. 41, pp. 33–38, July 2008.

[6] W. Song, S. Mukhopadhyay, and S. Yalamanchili, "Architectural reliability: Lifetime reliability characterization and management of many-core processors," *Computer Architecture Letters*, vol. PP, no. 99, pp. 1–1, 2014.

[7] B. Raghunathan, Y. Turakhia, S. Garg, and D. Marculescu, "Cherry-picking: Exploiting process variations in dark-silicon homogeneous chip multi-processors," in *Design, Automation Test in Europe Conference Exhibition (DATE), 2013*, pp. 39–44, March 2013.

[8] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Maestro: Orchestrating lifetime reliability in chip multiprocessors," in *Proceedings of the 5th International Conference on High Performance Embedded Architectures and Compilers*, HiPEAC'10, (Berlin, Heidelberg), pp. 186–200, Springer-Verlag, 2010.

[9] H. Kim, A. Vitkovskiy, P. V. Gratz, and V. Soteriou, "Use it or lose it: Wear-out and lifetime in future chip multiprocessors," in *Proceedings of the 46th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO-46, (New York, NY, USA), pp. 136–147, ACM, 2013.

[10] 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 *Proceedings of the 51st Annual Design Automation Conference*, DAC '14, (New York, NY, USA), pp. 170:1–170:6, ACM, 2014.

[11] T. E. Carlson, W. Heirman, and L. Eeckhout, "Sniper: Exploring the level of abstraction for scalable and accurate parallel multi-core simulations," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, pp. 52:1–52:12, Nov. 2011.

[12] "HotSpot Program." http://lava.cs.virginia.edu/HotSpot/versions.htm.

[13] J. R. Black, "Electromigration-a brief survey and some recent results," *IEEE Transactions on Electron Devices*, vol. 16, no. 4, pp. 338–347, 1969.

[14] I. Blech, "Electromigration in thin aluminum films on titanium nitride," *Journal of Applied Physics*, vol. 47, no. 4, pp. 1203–1208, 1976.

[15] 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.

[16] V. Sukharev, "Beyond Black's equation full-chip EM/SM assessment in 3D IC stack," *Microelectronic Engineering*, vol. 120, pp. 99–105, May 2014.

[17] 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.

[18] S. Chatterjee, M. Fawaz, and N. F. Najm, "Redundancy-Aware Electromigration Checking for Mesh Power Grids," in *Proc. Int. Conf. on Computer Aided Design (ICCAD)*, 2013.

[19] 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. on Computer Aided Design (ICCAD)*, pp. 327–334, IEEE, November 2004.

[20] S. Wang and J.-J. Chen, "Thermal-aware lifetime reliability in multicore systems," in *Quality Electronic Design (ISQED), 2010 11th International Symposium on*, pp. 399–405, March 2010.

[21] A. Das, A. Kumar, and B. Veeravalli, "Reliability-driven task mapping for lifetime extension of networks-on-chip based multiprocessor systems," in *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, (San Jose, CA, USA), pp. 689–694, EDA Consortium, 2013.

[22] C. Watkins and P. Dayan, "Q-learning," *Machine Learning*, vol. 8, no. 3-4, pp. 279–292, 1992.

[23] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 1st ed., 1998.

[24] T. Jaakkola, M. I. Jordan, and S. P. Singh, "On the convergence of stochastic iterative dynamic programming algorithms," *Neural Computation*, vol. 6, pp. 1185–1201, Nov. 1994.

[25] H. Shen, J. Lu, and Q. Qiu, "Learning based dvfs for simultaneous temperature, performance and energy management," in *Quality Electronic Design (ISQED), 2012 13th International Symposium on*, pp. 747–754, March 2012.

[26] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, pp. 1–7, Aug. 2011.

[27] M. M. K. Martin, D. J. Sorin, B. M. Beckmann, M. R. Marty, M. Xu, A. R. Alameldeen, K. E. Moore, M. D. Hill, and D. A. Wood, "Multifacet's general execution-driven multiprocessor simulator (gems) toolset," *SIGARCH Comput. Archit. News*, vol. 33, pp. 92–99, Nov. 2005.

[28] K. Ghose and et al, "Marssx86: Micro architectural systems simulators," in *ISCA Tutorial Session*, 2012.

[29] 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, pp. 18–31, July 2006.

[30] J. H. Ahn, S. Li, O. Seongil, and N. Jouppi, "Mcsima+: A manycore simulator with application-level+ simulation and detailed microarchitecture modeling," in *Performance Analysis of Systems and Software (ISPASS), 2013 IEEE International Symposium on*, pp. 74–85, April 2013.

[31] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques*, PACT '08, (New York, NY, USA), pp. 72–81, ACM, 2008.

[32] S. Woo, M. Ohara, E. Torrie, J. Singh, and A. Gupta, "The splash-2 programs: characterization and methodological considerations," in *Computer Architecture, 1995. Proceedings., 22nd Annual International Symposium on*, pp. 24–36, June 1995.

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

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

[35] T. Kolpe, A. Zhai, and S. Sapatnekar, "Enabling improved power management in multicore processors through clustered dvfs," in *Proc. Design, Automation and Test In Europe. (DATE)*, pp. 1–6, March 2011.