

Resilient Dynamic Power Management under Uncertainty

Hwisung Jung and Massoud Pedram

Department of Electrical Engineering, University of Southern California

Los Angeles, CA 90089

{hwijung, pedram}@usc.edu

Abstract

With the increasing levels of variability and randomness in the characteristics and behavior of manufactured nanoscale structures and devices, achieving performance optimization under process, voltage, and temperature (PVT) variations as well as current, voltage, and thermal (CVT) stress has become a daunting, yet vital, task. In this paper, we present a stochastic dynamic power management (DPM) framework to improve the accuracy of decision making under probabilistic conditions induced by PVT variations and/or stress. More precisely, we propose a resilient power management technique that guarantees to select an optimal policy under sources of uncertainty. A key characteristic of the proposed technique is that the effects of uncertainties due to variability and stress are captured by stochastic processes which control a self-improving power manager. Simulation results with a 65nm processor design show that, compared to the worst-case PVT conditions, the proposed DPM technique ensures energy efficiency, while reducing the uncertain behaviors of the system.

1. Introduction

As nanoscale VLSI circuits are becoming sensitive to the rising levels of variability in process and design parameters, guaranteeing the quality of system-level performance optimization techniques is becoming of great concern. Within-chip variations are typically passed into the delay budget of each circuit [1]. However, the worst-case behavior of the circuit (e.g., critical path delay) does not always correspond to the combination of worst-case points of individual parameters (e.g., load capacitance, intrinsic delay, and slew rate). Furthermore, a lot of silicon performance is left untapped under the worst-case assumption. IC designers can no longer afford to mislay performance due to unacceptable levels of inaccuracy in their estimation/modeling techniques [2]. Thus, it is important to do rigorous modeling of variability early in the design cycle. At the same time, it is turning out to be increasingly difficult to meet an acceptable safety margin for manufactured IC's because the device and interconnection aging and/or failure processes (which are in turn due to current, voltage and thermal stress) are becoming more severe as we scale down the devices and critical dimensions on the VLSI chips. For example, under normal operation conditions, the transistor characteristics can change by more than 10% over a 10-year period. Furthermore, stress, e.g., hot carrier injection (HCI) in NMOS transistors, negative bias temperature instability (NBTI) in PMOS transistors, and time-dependent dielectric breakdown (TDDB) change the threshold voltage of transistors so as to slow down the switching speeds [3]. There are also interconnect aging effects such as electro-migration.

The semiconductor industry has begun to define the lifetime of IC's as the duration of time after which 0.1% of the manufactured IC's will fail. This is clearly a much more stringent requirement than the traditional mean time to failure (MTTF), which means the

average life function of an IC. Note that MTTF is equal to the duration of time after which 50% of IC's fail only if the life time probability distribution function is symmetric, which is generally not the case. In general, the reliability of an IC should be specified as a percentage value with an associated time. Ideally, a confidence level should also be given, which allows for consideration of the variability of data with respect to the specification.

Increasing interest has been given to the problem of modeling and reducing variability in the design parameters. The work in [1] studies the parameter variations in nanometer and their impacts on leakage reduction techniques for a microprocessor. By emphasizing that the leakage is critically dependent on the operating temperature and power supply, the authors in [4] present a variation resilient circuit design technique for maintaining parametric yield of design under inherent variation in process parameters. A full chip leakage estimation technique under variability is presented in [5] to account for power supply and temperature variations. In [6], the impacts of threshold voltage variations on the leakage power are modeled in a probabilistic way, where these models are subsequently employed to minimize the leakage power dissipation, while satisfying certain performance requirements. The authors in [7] show that interactions between voltage, frequency, and temperature significantly impact the energy-delay-product of a target system. The work in [8] presents an analog circuit aging simulation technique based on a behavioral model which includes the effects of degradations on circuit parameters.

Most of the previous work has focused on the variability modeling, analysis, and control at the lower levels of design abstraction, e.g., by using physical design optimization and/or logic synthesis. Although it is prudent to account for various sources of variability earlier in the design process when developing resource management (e.g., DPM), variations at the higher levels of design abstraction are often translated into uncertainty because the underlying RT-level/physical realization is not available. At the same time, observations made about the current state of the system tend to be imperfect, and approximate, which in turn gives rise to uncertainty about the system state.

In this paper, we attempt to address uncertainty management issues in performance optimization at the system-level by ensuring IC designers' goal to produce low-power design with reliability. We propose a resilient power management framework which handles the parameter variations during power management. Our proposed DPM framework is based on stochastic (but stationary) processes that combine the dynamic programming (DP) and the expectation-maximization (EM) algorithms, which enable a power manager to predict uncertain state of the system in a dynamic environment. Traditional DPM approaches [9], which define a power manager to interact with the system resources through its commands and their associated costs, tend to be less than satisfactory in the presence of variability. This is because they assume different variables of the system are (i) directly observable and (ii) deterministic. Key feature of the proposed technique is to provide resiliency to the power manager to deal with various source of uncertainty.

¹ This research is supported in part by the National Science Foundation under grant no. 0509564.

The remainder of this paper is organized as follows. Section 2 provides some background of paper. The details of the proposed DPM framework are given in section 3. Section 4 presents a proposed DPM technique. Experimental results and conclusions are given in section 5 and section 6.

2. Background

As process geometries shrink, the leakage power dissipation is becoming an important portion of total power of an electronic system, where the prominence of leakage current in ICs has been spurred by the continued scaling of subthreshold voltages and gate oxide thickness. Both subthreshold and gate leakage currents are known to be highly sensitive to process variations due to their exponential dependence on many key process parameters [1][2]. Furthermore, with regard to environmental variations, it is important to implement ICs that have well-distributed temperature profiles, and to carefully design supply networks to provide robust supply voltage and ground levels throughout the chip. Figure 1 shows an example of leakage power variation on a 32bit RISC processor that we designed with 65nm technology, where the leakage power compromises integration density, packaging density, and reliability.

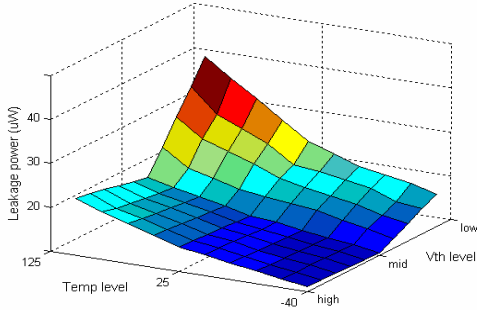


Figure 1. Leakage power for different levels of variability.

In general, modern transistors are stressed by high electric field, high temperature, and high switching activity over extended periods of time. These stress factors lead to device aging, resulting in performance degradation and eventual design failure during the expected lifetime of the designs. The MOS transistor aging phenomena, such as NBTI (Negative Bias Temperature Instability), HCI (Hot Carrier Injection), and TDDB (Time Dependent Dielectric Breakdown), are nowadays the most critical device degradation mechanisms. They are fast becoming a limiting factor in scaling of modern CMOS technologies. For example, NBTI, which tends to significantly increase the threshold voltage, exhibits wide variations from one wafer run to next and thereby reduces the safety margin between actual lifetime of MOSFETs and the expected and/or required product lifetime. Note that the NBTI effect gets worse at higher temperature. In addition, the HCI effect, whereby carriers injected into the gate oxide near the drain end of a strongly inverted channel add to the trapped ion density at the silicon oxide and substrate interface and thus increase the threshold voltage of an NMOS transistor, also exhibits wide variations, especially resulting in asymmetrical device characteristics after stress. Contrary to NBTI, however, HCI gets worse at lower temperature [11]. Further details about these phenomena are omitted here to save space. Interested reader may refer to [11].

The above-mentioned MOS aging phenomena influence the performance of the circuit due to corresponding device parameter degradation. So far, technology developers have assumed the key responsibility to adjust the process technology features and device reliability characteristics is to achieve the required lifetime for the

microelectronic circuits. However, the circuit reliability can no longer be the task of process and device engineers alone. Although performance analysis tools (e.g., PrimeTime [12]) provide reliable bounds on the delay of circuits, they cannot properly account for the variability and/or stress inherent in semiconductor process. For example, Figure 2 illustrates variational effect in gate-level static timing analysis, where gate delays are calculated based on lookup tables. Every point in the table represents characterized spice timing for cell given particular input transitions and output capacitance. Obviously, not all possible input transitions and output capacitance values for a given cell can be characterized. In this figure, the closest four characterized points in the table are used to interpolate them for calculating the delay. Thus, although these analysis tools provide the estimate of performance parameters during the design time, they cannot guarantee that the resulting performance is accurate after fabrication.

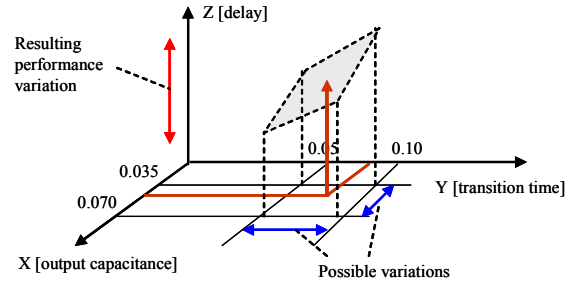


Figure 2. Variational effect on timing delay.

3. Resilient Decision Making

In this section, we present a theoretical framework to construct a resilient power management process under uncertainty induced by variability.

3.1 Partially Observable Markov Decision Process

To capture the impact of variability induced by PVT variations and/or stress effects which often result in the uncertainty in the performance parameters of interest, the observation can be used in a stochastic manner to identify the performance (e.g., power dissipation) of the system. In other words, power management can be performed under a stochastic environment whose performance state is partially observable due to variability. When designing a power manager that acts under uncertainty, it is convenient to model the power manager in the context of partially observable Markov decision process (POMDP) [13], which provides a way to model uncertainty in a real world system's actions and perceptions.

A POMDP is a special Markov Decision Process (MDP) where the state is only partially observable. A POMDP is represented by a tuple (S, A, O, T, Z, c) where state space S comprises of a finite set of (nominal) states, action space A consists of a finite set of actions, and observation space O contains a finite set of observations. T is a state transition probability function, Z is an observation function, and c is a one-step cost function. The state transition function determines the probability of a transition from a state s to another state s' after executing an action a , i.e., $T(s', a, s) = \text{Prob}(s^{t+1} = s' | a^t = a, s^t = s)$.² The observation function, which captures the relationship between the actual state and the observation, is defined as the probability of making observation o' after taking an action a that would land the system in state s' , i.e., $Z(o', s', a) = \text{Prob}(o^{t+1} = o' | a^t = a, s^{t+1} = s')$. We consider a cost function which assigns a value to each state and action pair by adopting a conventional

² In this paper, subscripts denote the state information whereas superscripts denote the time stamp.

approach, whereby an immediate cost, $c(s, a)$, is incurred when action a is chosen in state s . Note that the costs can be set by the applications or the developers.

Instead of making decisions based on the current perceived state of the system, the POMDP maintains a *belief*, i.e., a probability distribution over the possible (nominal) states of the system, and makes decisions based on its current belief. The *belief state* at time t is a $|S| \times 1$ vector of probabilities defined as: $b^t := [b^t(s)]$, $\forall s \in S$, where $b^t(s)$ is the posterior probability distribution of state s at time t . Note that $\sum_{s \in S} b^t(s) = 1$. Based on the belief state, an action a^t is chosen from a set of available actions. A *policy* is defined as a sequence of mappings from the belief states to actions $\pi = \{\pi^t\}$.

3.2 Rationale for POMDP-based Power Management

It is useful to describe how the POMDP can be adapted to the power management under uncertainty. In this paper, the POMDP is used to model the decision-making problem in a partially observable environment. Figure 3 shows the high-level structure of the POMDP-based power manager. The proposed power manager interacts with an uncertain environment (which is affected by PVT variations and/or stress effects) and tries to minimize its long term cost by choosing appropriate actions (i.e., voltage-frequency values). The actions commanded by the power manager change the performance state of the system and lead to quantifiable rewards/penalties. In this paper, we consider the case where actions incur a cost (i.e., energy dissipation), where the power manager's goal is to devise a policy, which minimizes the total expected energy dissipation.

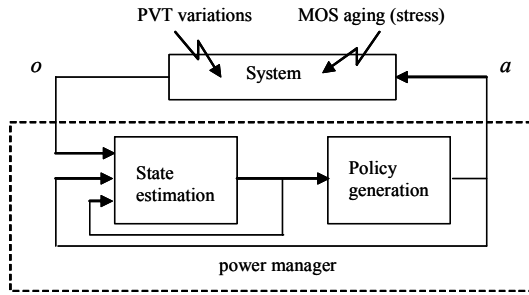


Figure 3. Structure of the proposed power manager.

In our formulation of the decision-making strategy, we define state $s \in S$ as the dissipated power level, where S is the set of available system states. Furthermore, we use an *observation* (i.e., a temperature measurement) to help identify the system state. We assume that multiple on-chip thermal sensors provide information about the temperatures in different zones of the chip [14].

As shown in Figure 3, the power manager consists of two functional components. The first component is the state estimation based on the expectation-maximization (EM) algorithm, which predicts the system state s of the system, and the second component is a power management policy generation, which assigns optimal actions to the system.

3.3 Power Management Framework

In a partially observable environment, the belief state is a vector of the posterior probability distribution of state s . For example, assume that we define three nominal states, e.g., $s_1 = [0.5 \ 0.8]$, $s_2 = [0.8 \ 1.1]$, and $s_3 = [1.1 \ 1.4]$ in terms of the range of power consumption values (in W). Now, assume the belief state $b^t(s)$ is $[b(s_1) \ b(s_2) \ b(s_3)] = [0.1 \ 0.7 \ 0.2]$; the probability of being in state s_2 is 0.7, which means that s_2 is the most probable state of the system at time t . Figure 4 (a) illustrates a partial observation (i.e., temperature measurement) and its effect on the probability density function.

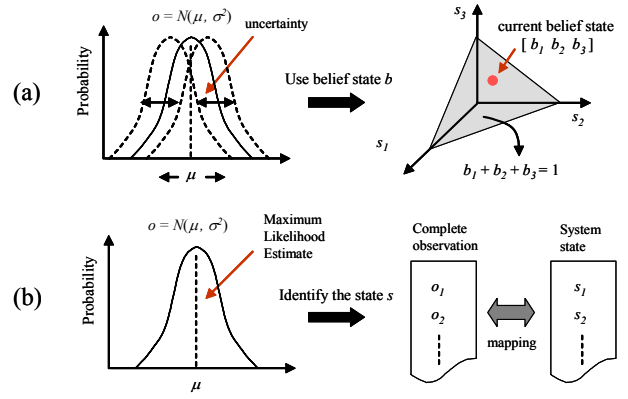


Figure 4. (a) Effect of hidden data on the pdf of the measured data and the concept of a belief state, and (b) Use of the ME algorithm to estimate the most probable system state without having to resort to a belief state representation.

If we are given a belief state b^t for time t and we perform an action a and get observation o^t , we can compute the successor belief state for time $t+1$ as follows:

$$b^{t+1}(s') = \frac{Z(o^t, s', a) \sum_s b^t(s) T(s', a, s)}{\sum_{s, s''} Z(o^t, s'', a) b^t(s) T(s'', a, s)} \quad (1)$$

A key result is that if we maintain the belief state and update it in according to Eqn. (1), then using this belief state will give us just as much information as the entire action-observation history. This shows that the optimal POMDP solution is Markovian over the belief state space. In fact, using the belief space as our set of states, we can use the transition and observation probabilities to create a completely observable, regular (albeit continuous state space) MDP that is equivalent to the original POMDP [15]. Note that the complexity of computation required by Eqn. (1) for updating the belief state grows rapidly with the number of state variables, making it infeasible for real-time applications, e.g., online DPM techniques. In general, finding policies for POMDP models is known to be computationally intractable. In particular, calculating exact solutions for the finite-horizon stochastic POMDP problems is P-SPACE hard [16]. Therefore, exact solutions cannot be found for POMDPs with more than a handful of states. Indeed, solving a POMDP problem (i.e., obtaining an optimal policy while computing the belief states) is extremely expensive because of the complexity of calculating the exact belief state [17]. To overcome this difficulty, one is usually forced to estimate the system state by some other approach. By doing so, the overwhelming complexity in deriving a power management policy for every possible situation is avoided.

We utilize a system state estimation technique based on the “expectation-maximization” (EM) algorithm [18], where the observations can be viewed as incomplete data. The EM algorithm is a general technique which can be used to determine the maximum likelihood estimate (MLE) of the parameters of an underlying distribution from some given data when the observed data is incomplete [19]. Alternatively stated, it is used to find the MLE of some parameters in a probabilistic model, where the model depends on unobserved latent (hidden) variables. The main advantage of the EM in our problem setup is that the EM deals with uncertain observation when computing the MLE.

Let o denote observed data (measurement) and let m denote the missing data. Together o and m form the *complete data*. Notice that m can be an actual missing measurement, or in our problem context,

a hidden source of variation that affects the measurement. Let p denote the joint probability density function of the complete data with parameters given by vector θ , i.e., we have $p(o, m | \theta)$. (θ may for example correspond to the mean value and variance of a Gaussian distribution.) This function can also be considered as the complete data likelihood, that is, it can be thought of as a function of θ and expressed as:

$$p(o, m | \theta) = p(m | o, \theta)p(o | \theta) \quad (2)$$

by using the Bayes rule.

The EM algorithm iteratively improves an initial estimate θ^0 by constructing new estimates θ^1, θ^2 , and so on. Note that an individual re-estimation step that derives θ^{n+1} from θ^n takes the following form:

$$\theta^{n+1} = \arg \max_{\theta} Q(\theta) \quad (3)$$

where $Q(\theta)$ is the expected value of the log-likelihood of complete data. Since we do not know the complete data, we cannot determine the exact value of the likelihood, but given the observed data o , we can calculate a posteriori estimates of the probabilities for the various values of m . For each set of m values, there is a likelihood value for θ , and we can hence calculate an expected value of the likelihood with the given values of o 's. In our problem setup, Q is given by:

$$Q(\theta) = E_m (\log p(o, m | \theta) | o) \quad (4)$$

where it is understood that this denotes the conditional expectation of $\log p(o, m | \theta)$ being taken with the θ used in $p(m | o, \theta)$ fixed at θ^n . In other words, θ^{n+1} is the value that *maximizes* the conditional *expectation* of log-likelihood of the complete data given the observed variables under the previous parameter value. The expectation $Q(\theta)$ may be written as:

$$Q(\theta) = \int_{-\infty}^{\infty} p(m | o) \log p(o, m | \theta) dm \quad (5)$$

These two steps (Expectation and Maximization) are repeated until the convergence of the parameters $|\theta^{n+1} - \theta^n| \leq \omega$, where the value of ω is selected by system developers [21]. It can be shown that the EM iteration does not decrease the observed data likelihood function. However, there is no guarantee that the sequence converges to a maximum likelihood estimator. There are a variety of heuristic approaches for escaping a local maximum such as using different random initial estimates or applying simulated annealing.

The EM algorithm finds θ that maximizes the complete-data likelihood, which in turn removes the effect of hidden variables and allows us to calculate the MLE of the system state without having to resort to the belief state representation (cf. Figure 4 (b)).

4. Resilient DPM

In this section, we present a dynamic power management technique which comprises of a state estimation and a policy generation step.

4.1 EM-based State Estimation

The state estimation step corresponds to the estimation of the complete observation (o, m) , where we use the expectation-maximization (EM) algorithm. From this information, we can identify the system state s from the complete data through the pre-defined *observation-state mapping table*. Note that this mapping table, which indicates the performance-state of the system based on complete observation, is obtained by simulations during design time.

Figure 5 summarizes the flow of the state estimation, where we first set the initial value for vector θ .

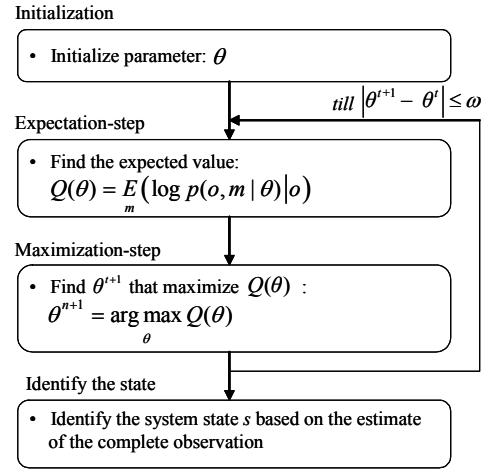


Figure 5. The flow of the state estimation by the EM algorithm.

Considering algorithms for state estimation, there are a number of other methods for estimation such as moving average filter [10], least mean square filter [22], and Kalman filter [23]. In our problem setup, we have found that the EM algorithm is more efficient than other methods since it provides a parameterized class of density functions, which minimizes the computational complexity.

4.2 Policy Generation

The policy generation step deals with the value function to determine the set of optimal actions. Note that a policy is defined as a sequence of mappings from states to actions.

We develop a policy generation technique by using well-known dynamic programming method used to solve problem which exhibits the properties of optimal cost. We speak of the minimum cost of a system state which is the expected infinite discounted sum of cost that the system will accrue if it starts in that state and executes the optimal policy. Generally, using π as a complete decision policy, this minimum cost is written as

$$\Psi^*(s) = \min_{\pi} E \left(\sum_{t=0}^{\infty} \gamma^t \cdot c(t) \right) \quad (6)$$

where γ is a discount factor, $0 \leq \gamma < 1$, and $c(t)$ is the cost at time t .

In our problem setup, the minimum cost function is unique and can be defined

$$\Psi^*(s) = \min_a \left(C(s, a) + \gamma \sum_{s' \in S} T(s', a, s) \Psi^*(s') \right) \quad \forall s \in S \quad (7)$$

which assert that the cost of a state s is the expected immediate cost plus the expected discounted cost of the next state, using the best available action. Here, the immediate costs are defined as the power-delay product (PDP), i.e., average energy of the system, which is incurred due to each state-action pair. By applying Bellman's principle of optimality [24], given the optimal cost function, we can specify the optimal policy as

$$\pi^*(s) = \arg \min_a \left(C(s, a) + \gamma \sum_{s' \in S} T(s', a, s) \Psi^*(s') \right) \quad (8)$$

Simply stated, the power manager determines the optimal action based on Eqn. (8) while utilizing the EM algorithm to estimate the system state. The power manager observes the performance of the system at each event (e.g., time-based or interrupt-based) occurrence. These events occur at so-called decision epochs. The task of casting the decision epochs to absolute time units is achieved by the system developer. Note that we focus on battery operated systems that strive to conserve energy to extend the battery

life, unlike AC line powered high performance systems that deliver maximum performance while guaranteeing performance constraints.

Given $C(s, a)$ and $T(s', a, s)$, one way to find an optimal policy is to find the minimum cost function. It can be determined by an iterative algorithm (cf. Figure 6) called value iteration [25] that can be shown to converge to the correct Ψ^* values. It is not obvious when to stop this algorithm. One important result bounds the performance of the current greedy policy as a function of the Bellman residual of the current cost function [26]. It states that if the maximum difference between two successive cost functions is less than ϵ , then the cost of the greedy policy (i.e., the policy obtained by choosing, in every state, the action that minimizes the estimated discounted cost, using the current estimate of the cost function) differs from the cost function of the optimal policy by no more than $2\epsilon\gamma / (1-\gamma)$ at any state. This provides a stopping criterion for the algorithm.

```

1: initialize  $\Psi(s)$  arbitrarily
2:   loop until policy good enough
3:     loop for  $\forall s \in \mathcal{S}$ 
4:       loop for  $\forall a \in A$ 
5:          $Q(s, a) = C(s, a) + \gamma \sum_{s' \in \mathcal{S}} T(s', a, s) \Psi(s')$ 
6:          $\Psi(s) = \min_a Q(s, a)$ 
7:       end loop
8:     end loop
9:   end loop

```

Figure 6. The value iteration algorithm.

5. Experimental Results

In the experimental setup, we applied the proposed DPM technique to a 32bit MIPS-compatible processor, which has 5-stages pipeline, instruction/data caches, and internal SRAM for code/data storage. To apply the effect of variability in this processor, we relied on the detailed RTL of the processor synthesized with TSMC 65nmLP library, where power numbers (i.e., leakage and dynamic power) are achieved through the Power Compiler [12] with the exact switching activity information.

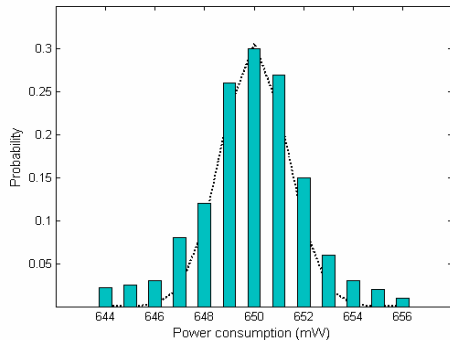


Figure 7. Probability density function for power dissipation.

In the first experiment, we analyze the possible variations of the processor in terms of power dissipation. By varying process corners during the simulation setup, we achieve power numbers for the processor after running the tasks. As an application, we perform real-time TCP/IP-related tasks (i.e., TCP segmentation and checksum offloading [27]) at the designed processor. After running a number of simulations, we achieve the probability density function for the total power consumption of the processor as shown in Figure 7, where the mean value is 650mW and variance σ^2 is 3.1 (i.e., $N(650, 3.1)$). Next we calculate the on-chip temperature (i.e., observation) based on the measured power, which is affected by

sources of variability. In doing so, we rely on the temperature calculation equations of [28]. Note that the temperature values ($^{\circ}\text{C}$) are obtained during the active state of the processor. We use the extracted thermal data [29] for PBGA as reported in Table 1, where T_A is the ambient temperature, θ_{JA} is the thermal resistance for junction-to-ambient, and ψ_{JT} denotes the junction-to-top of package thermal characterization parameter.

Table 1. Package thermal performance data ($T_A = 70^{\circ}\text{C}$).

Air velocity		$T_{J,max}$ [$^{\circ}\text{C}$]	$T_{T,max}$ [$^{\circ}\text{C}$]	ψ_{JT} [$^{\circ}\text{C}/\text{W}$]	θ_{JA} [$^{\circ}\text{C}/\text{W}$]
m/s	ft/min				
0.51	100	107.9	106.7	0.51	16.12
1.02	200	105.3	104.1	0.53	15.62
2.03	300	102.7	101.2	0.65	14.21

The second experiment is designed to demonstrate the effectiveness of the proposed DPM framework. We first set the parameter values for the simulation setup as shown in Table 2, where we have a set of three actions $\{a_1, a_2, a_3\}$, three states $\{s_1, s_2, s_3\}$, and three observations $\{o_1, o_2, o_3\}$. Note that the actions are defined as: $a_1 = [1.08\text{V} / 150\text{MHz}]$, $a_2 = [1.20\text{V} / 200\text{MHz}]$, and $a_3 = [1.29\text{V} / 250\text{MHz}]$. The state set $\{s_1, s_2, s_3\}$ of the processor may be defined as the range of power dissipation, where we achieve power number through simulations. A set of cost values is defined as the power-delay product (PDP) of the processor, where the normalized PDP is achieved by multiplying the average power consumption by the average execution delay. For example, the cost values for each state when action a_1 is defined as $c(s_1, a_1)$, $c(s_2, a_1)$, and $c(s_3, a_1)$, which are 541, 500, and 470.

Table 2. The parameter values for a given experiment.

State	Description [W]	Observation	Description [$^{\circ}\text{C}$]	cost $c(s, a)$		
				s_1	s_2	s_3
s_1	[0.5 0.8]	o_1	[75 83]	a_1	[541 500 470]	
s_2	(0.8 1.1)	o_2	(83 88]	a_2	[465 423 381]	
s_3	(1.1 1.4]	o_3	(88 95]	a_3	[450 508 550]	

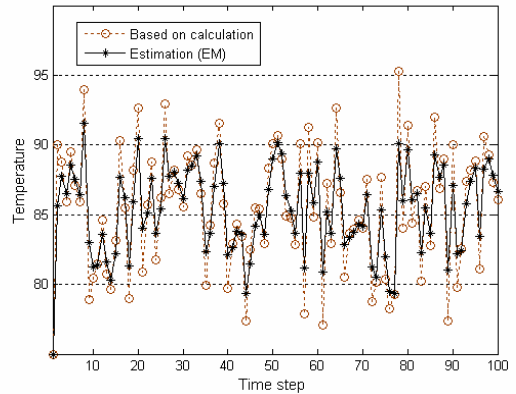


Figure 8. Trace of temperatures from the thermal calculator and from ML estimates.

Figure 8 shows the traces of on-chip temperature from temperature calculation equations based on power measurements and that from MLE. Because we do not have a packaged IC equipped with a thermal sensor to report the on-chip temperature, we estimate the on-chip temperature by utilizing $T_{chip} = T_A + P \cdot (\theta_{JA} - \psi_{JT})$ based on the parameter values provided in Table 1. In this equation, P is the simulated power dissipation value. For this experiment, we initially set $\theta^0 = (70, 0)$, where 70°C is the initial

most probable die temperature. Note that time steps are abstractly defined and the power manager issues a command at each time step. The results indicate that the proposed estimation exhibits great accuracy, where the estimation error is on average less than 2.5°C.

Simulation results in Figure 9 assess the effectiveness of the policy generation algorithm in choosing the optimal action based on the information provided in Table 1. We set the discount factor as 0.5 to evaluate the value function. It is seen that an optimal action is chosen to minimize the value function. Note that the conditional transition probabilities are given in advance, where extensive offline simulations are used to achieve the values of probabilities.

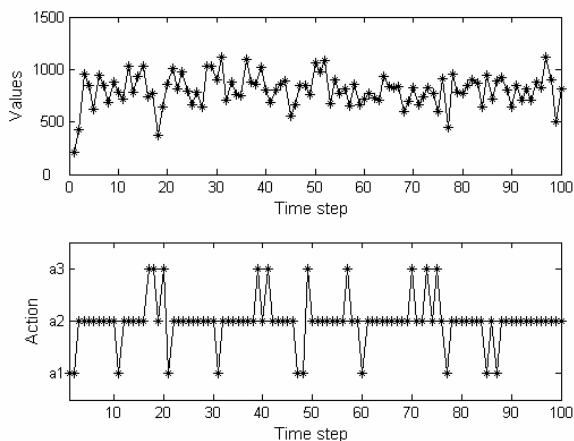


Figure 9. Evaluation of policy generation algorithms.

In the third experiment, we compare the proposed uncertainty-aware DPM technique with conventional DPM techniques operating under two different conditions, i.e., worst and best corner cases in a 65nm CMOS process technology. The purpose of this simulation is to investigate how robustly the proposed approach can handle variability during the power management process. The optimal DPM policy is achieved by evaluating the value function with the derived state transition probabilities, as illustrated in Figure 8. In our approach, we performed the tasks while varying the operating conditions, and identified the system state by using the EM algorithm to determine the MLE of the system state given noisy temperature observations. Table 3 summarizes these simulation results in terms of power, energy (normalized), and energy-delay-product (EDP) as the figure of merit. Energy and EDP results in columns 5 and 6 are normalized to the best case results. Clearly, the uncertainty-aware DPM approach cannot do any better than a conventional DPM at the best corner case. The expectation, however, is that it will outperform the conventional DPM at the worst corner case, while ensuring energy efficiency. The results of Table 3 confirm this expectation.

Table 3. Comparing results of our approach with the corner-based results.

	Minimum Power	Maximum power	Average Power	Energy (normalized)	EDP (normalized)
Our approach	0.71W	1.12W	0.97W	1.14	1.34
Worst case	0.77W	1.26W	1.02W	1.47	2.30
Best case	0.96W	1.31W	1.15W	1.00	1.00

6. Conclusion

We described a resilient DPM technique which guarantees to select an optimal power management policy under variability. The proposed DPM framework, which brings PVT variational and/or stress effects to the forefront of decision-making strategy, controls

the uncertain behavior of the system, ensuring energy efficiency. In the experimental results included in the paper, we have shown that our mathematical framework can handle system's behavior under uncertain information. Being able to handle various sources of uncertainty improves the accuracy and robustness of the design.

References

- [1] S. Borkar, et al., "Parameter Variations and Impact on Circuits and Microarchitecture," *Proc. of DAC*, Jun., 2003.
- [2] A. Srivastava, D. Sylvester, and D. Blaauw, *Statistical Analysis and Optimization for VLSI: Timing and Power*, Springer, 2005.
- [3] A. B. Kahng, "Design Challenges at 65nm and Beyond," *Proc. of DATE*, Mar., 2007.
- [4] K. Kang, K. Kim, and K. Roy, "Variation Resilient Low-Power Circuit Design Methodology using On-Chip Phase Locked Loop," *Proc. of DAC*, Jun., 2007.
- [5] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full Chip Leakage Estimation Considering Power Supply and Temperature Variations," *Proc. of ISLPED*, Aug., 2003.
- [6] M. Lie, W.S. Wang, and M. Orshansky, "Leakage Power Reduction by Dual-Vth Designs Under Probabilistic Analysis of Vth Variation," *Proc. of ISLPED*, Aug., 2004.
- [7] A. Basu, et al., "Simultaneous Optimization of Supply and Threshold Voltages for Low-Power and High-Performance Circuits in the Leakage Dominant Era," *Proc. of DAC*, Jun., 2004.
- [8] F. Marc, et al., "Improvement of Aging Simulation of Electronic Circuits Using Behavioral Modeling" *IEEE Trans. on Device and Materials Reliability*, Vol. 6, No. 2, Jun., 2006.
- [9] L. Benini, and G. De Micheli, *Dynamic Power Management: Design Techniques and CAD Tools*, Kluwer Academic Publishers, 1998..
- [10] A. Gosavi, *Simulation-Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*, Kluwer Publishers, 2003.
- [11] M. A. Alam, "A simple view of a complex phenomena," *Proc. of Int'l Reliability Physics Symposium*, Mar., 2006.
- [12] Synopsys PrimeTime / Power Compiler. <http://www.synopsys.com>
- [13] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Publisher, New York, 1994.
- [14] M. R. Stan, and K. Skadron, "Power-Aware Computing," *IEEE Computer*, Vol. 36, No. 1, Jan., 2003.
- [15] C. T. Striebel, "Sufficient statistics in the optimal control of stochastic systems," *J. Math. Anal. Appl.*, 12, pp. 576-592, 1965.
- [16] D. Burago, M. de Rougemont, and A. Slissenko, "On the complexity of partially observed Markov decision processes," *Theoretical Computer Science*, 157(2):161-183, 1996.
- [17] S. Paquet, G. Gordon, and S. Thrun, "Point-based Value Iteration: An Anytime Algorithm for POMDPs," *Proc. of Conf. on AI*, Aug., 2003.
- [18] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, 39(1):1-38, 1977.
- [19] O. Cappe, et al., "An Algorithm for Maximum Likelihood Estimation of Hidden Markov Models with Unknown State-Typing," *IEEE Trans. on Speech and Audio Processing*, Vol. 6, No. 1, Jan., 1998.
- [20] G. McLachlan, and T. Krishnan, *The EM Algorithm and Extensions*, Wiley-Interscience publisher, 1996.
- [21] J. A. Bilmes, "A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Model," *Technical Report*, TR-97-021, U.C. Berkeley, 1998.
- [22] P. Diniz, *Adaptive Filtering Algorithms and Practical Implementation*, Kluwer Academic, 1997.
- [23] R. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Journal of Basic Engineering*, Vol. 82, Series D, 1960.
- [24] R.E. Bellman, *Dynamic Programming*. Princeton Univ Press, 1957.
- [25] D. P. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models*. Prentice-Hall, Englewood Cliffs, NJ, 1987.
- [26] R. Williams, and L. Baird, "Tight Performance Bounds on Greedy Policies based on Imperfect Value Functions," Technical report NU-CCS-93-14, Northeastern University, Nov., 1993.
- [27] IEEE802.3 Ethernet document. <http://www.ieee802.org>
- [28] Y. Cheng, C. Tsai, C. Teng, and S. Kang, *Electrothermal Analysis of VLSI Systems*, Kluwer Academic Publishers, 2000.
- [29] Thermal data for MIPS processors. <http://www.broadcom.com>.