

A Stochastic Local Hot Spot Alerting Technique¹

Hwisung Jung

Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
e-mail: hwijung@usc.edu

Massoud Pedram

Dept. of Electrical Engineering
University of Southern California
Los Angeles, CA 90089
e-mail: pedram@usc.edu

Abstract - With the increasing levels of variability in the behavior of manufactured nano-scale devices and dramatic changes in the power density on a chip, timely identification of hot spots on a chip has become a challenging task. This paper addresses the questions of how and when to identify and issue a hot spot alert. There are important questions since temperature reports by thermal sensors may be erroneous, noisy, or arrive too late to enable effective application of thermal management mechanisms to avoid chip failure. This paper thus presents a stochastic technique for identifying and reporting local hot spots under probabilistic conditions induced by uncertainty in the chip junction temperature and the system power state. More specifically, it introduces a stochastic framework for estimating the chip temperature and the power state of the system based on a combination of Kalman Filtering (KF) and Markovian Decision Process (MDP) model. Experimental results demonstrate the effectiveness of the framework and show that the proposed technique alerts about thermal threats accurately and in a timely fashion in spite of noisy or sometimes erroneous readings by the temperature sensor.

I. Introduction

With IC process geometries shrinking below 65nm technology and many applications requiring higher performance, thermal control is becoming a first-order concern for not only IC designers but also package, board, and system designers due to the increased power density as well as the increasing vulnerability of the system. For example, the gate oxide lifetime is highly dependent on the junction temperature of the IC, where elevated temperature is a major contributor to lower IC reliability. If heat is not removed at a rate equal to or greater than its rate of generation, the junction temperature will rise [1], which reduces the mean time to failure (MTTF) for the devices. Furthermore, local hot spots, which have much higher temperatures compared to the average die temperature, are becoming more prevalent in VLSI circuits. Thus, identifying and removing heat from these hot spots is a major task facing design engineers concerned with higher circuit reliability.

As reported in [2]-[7], the problem of thermal modeling and management has received a lot of attention. The work presented in [2] relies on a compact thermal model to achieve a temperature-aware design methodology. A thermal control mechanism used to cool the microprocessor's temperature has been derived in [3]. Predictive thermal management [4], which

exploits certain properties of multimedia applications, is an example of online strategies for thermal management. In [5][6], design guidelines for power and thermal management for high-performance microprocessors are provided. A summary of research that combines thermal management techniques and pitfalls is provided in [1][7].

Much of the past work has examined techniques for thermal modeling and management, but these techniques may be ineffective if the accuracy of identifying local hot spots is in question. This is because thermal models, based on equivalent circuit models, cannot adequately capture structures with complex shapes and boundary conditions, which in turn gives rise to uncertainty in identifying local hot spots. In particular, it is extremely difficult to obtain exact solution of the heat transfer equations that arise from realistic die conditions [8]. Furthermore, temperature sensors have difficulty measuring the actual peak power dissipation and the resulting peak temperature, which renders stochastic the problem of identifying local hot spots. To the best of our knowledge, no research work has been conducted on estimation of the local hot spots on a die which rigorously account for the uncertainty in temperature sensing.

In this paper, we present a stochastic hot spot estimation technique, which alerts against thermal problems. Uncertainties in the temperature measurements and power state identification are modeled by using stochastic processes. Our proposed framework is based on the Kalman filter (KF) algorithm and the Markovian decision process (MDP) model, which enable the framework to predict thermal behavior and power state of the system under variable, and uncertain, environmental conditions. Note that KF provides an estimation technique of the most probable state of a continuous-state system [10] while MDP is a theory of modeling sequential decision problem in a discrete-state system [9]. The key rationale for utilizing MDP and KF for hot spot estimation is to manage uncertainty, combining continuous thermal state and discrete power state estimations, respectively.

The remainder of this paper is organized as follows. Section II provides some preliminaries of the paper, while section III describes the details of uncertainty-aware estimation framework. Section IV presents a hot spot alerting algorithm. Experimental results and conclusions are given in section V and section VI.

II. Preliminaries

An integrated circuit (device) is typically allowed to operate when the ambient air temperature, T_A , surrounding the device package, is within the range of 0 °C to 70 °C [11]. The package can be characterized thermally by a thermal resistance. The

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

value of thermal resistance determines the temperature rise of the junction above a reference point by $\theta_{JX} = (T_J - T_X) / P$, where θ_{JX} is the thermal resistance from the device junction to the specific environment ($^{\circ}\text{C}/\text{W}$), T_J is the device junction temperature ($^{\circ}\text{C}$), T_X is the reference temperature for a specified environment ($^{\circ}\text{C}$), and P is the device power dissipation (W). If the reference temperature is denoted as T_A (i.e., the temperature of ambient), T_B (i.e., the temperature of PCB board), or T_C (i.e., the temperature of the case top), then the thermal resistances for junction-to-air, θ_{JA} , junction-to-board, θ_{JB} , and junction-to-case, θ_{JC} , may be calculated as

$$\theta_{JA} = \frac{T_J - T_A}{P}, \quad \theta_{JB} = \frac{T_J - T_B}{P}, \quad \theta_{JC} = \frac{T_J - T_C}{P} \quad (1)$$

Note that the thermal resistance is highly dependent on the environment surrounding the package. As illustrated in Figure 1, heat is dissipated from the die into the ambient primarily through either the package encase bottom surface or its top surface, where PBGA (Plastic Ball Grid Array) plus HS (Heat Spreader) package is used. The arrows in this figure indicate the direction of heat flow. The two heat dissipation paths are graphically plotted in Figure 2, where the package is thermally represented with a two-resistance model, one corresponding to the heat transfer resistance from the device junction to the package bottom surface, θ_{JB} , and the other corresponding to the heat transfer resistance from the device junction to the package top surface, θ_{JC} . The thermal resistances, external to the package, include θ_{BS} , θ_{BA} , and θ_{CA} , which are determined by the thermal design of the target system. For example, if there are no heat sinks attached to the package in the system, the surface-to-air thermal resistance, θ_{BA} and θ_{CA} , can be estimated from

$$\theta_{BA} = \theta_{CA} = \frac{1}{h_s A_s} \quad (2)$$

where h_s and A_s denote the heat transfer coefficient and exposed surface area of the package, respectively [12]. Note that θ_{BS} is the PCB board spreading thermal resistance, influenced by component population on board.

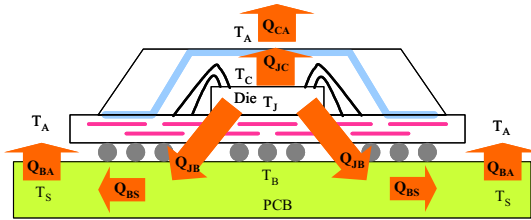


Figure 1. Heat flow in the Plastic Ball Grid Array plus Heat Spreader package.

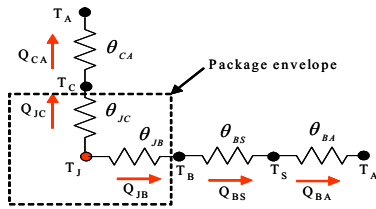


Figure 2. One of the IC package heat transfer paths and the corresponding thermal resistive model.²

² There is an analogy to Ohm's Law for electrical circuits where temperature replaces voltage and power dissipation replaces current.

Using the above-mentioned models, the package junction-to-air thermal resistance can be calculated from the following

$$\theta_{JA} = \left(\frac{1}{\theta_{JB} + \theta_{BS} + \theta_{BA}} + \frac{1}{\theta_{JC} + \theta_{CA}} \right)^{-1} \quad (3)$$

Then, the junction temperature can be estimated with

$$T_J = T_A + P \cdot \theta_{JA} \quad (4)$$

where the goal of thermal design of the package is to maintain the device θ_{JA} value small enough so that the junction temperature T_J does not exceed a maximum specified value during operation. It is worthwhile to note that θ_{JA} cannot be modeled directly due to the complexity of thermal models for the package, cooling system, and board stack-up. In addition, θ_{JA} is assumed to be a single parameter under the assumption that device power dissipation, P , is distributed uniformly across the die, which is not a realistic assumption. In practice, the package case top temperature, T_C , is utilized along with temperature measurements to estimate T_J .

Temperature reading can be performed by either external or internal thermal sensors. External thermal sensors, e.g., thermocouples, incur a rather large time delay in reading the temperature and tend to produce less accurate temperature measurements of T_J [1]. Internal thermal sensors, e.g., analog/digital CMOS sensors [13], which can be deployed in large numbers across a chip, have been widely used in pursuit of higher accuracy in measuring T_J . However there still remain inaccuracies associated with the internal sensors. For example, current biased temperature sensors are sensitive to noise on power and ground lines, and thus the sensor output for low temperature reading is affected by process variations [14], etc.

III. Estimation under Uncertainty

In this section, we present the theory of how to predict temperature under uncertainty (variability, noise, or error) in temperature reports.

A. Background on POMDP and Kalman filter

A Partially Observable Markov Decision Process (POMDP) [9] is a special Markov Decision Process (MDP), where the state is only partially observable. A POMDP, a way to model uncertainty in perceptions/observations made by a real system, is represented by a tuple (S, A, O, T, Z) where the state space S comprises of a finite set of states, the action space A consists of a finite set of actions, while the observation space O contains a finite set of observations. T is a state transition function, and Z is an observation 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., $T(s', a, s) = \text{Prob}(s^{t+1} = s' \mid a^t = a, s^t = s)$.³ 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., $Z(o', a, s') = \text{Prob}(o^{t+1} = o' \mid a^t = a, s^{t+1} = s')$. Since our goal in the problem setup is to find the best estimate of the current state, some elements of the POMDP model (i.e., reward function and policies) are not applicable to this paper.

Instead of making decisions based on the current perceived state of the system, the POMDP maintains a *belief*, i.e., a

³ In this paper, subscripts denote the state information whereas superscripts denote the time stamp.

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 . These probabilities are calculated as a function of the observed history. Note that $\sum_{s \in S} b^t(s) = 1$.

The Kalman filter (KF) [10] is designed to estimate the state of a discrete-time controlled process that is governed by the linear stochastic difference equation (e.g., Eqn. 6(a)) based on the previous state, previous action, and the current observation. Briefly, the KF operates by first predicting the state of the process using the state update equations, then correcting the prediction using an observation, where error covariances of the state and the observation are used to balance the prediction models. In dynamic environments, the KF dynamics results from the consecutive cycles of prediction and filtering, where in turn the KF finds the optimal estimate of the process state by minimizing the expected least-square error. The details of the POMDP models and KF algorithms are omitted here to save space. Interested readers may refer to [9][10].

B. Rationale for Developing Uncertainty Management

As illustrated in section 2, the junction temperature, T_j , is not estimated easily with Eqn. (4) due to the complexity of modeling θ_{JA} . To overcome this difficulty, we use an observation, i.e., temperature reading T_T of the package top obtained by a thermal sensor, to estimate T_j as follows:

$$T_j = T_T + \psi_{JT} P \quad (5)$$

where ψ_{JT} is the junction-to-top of package thermal characterization parameter used as a measure of the temperature difference between junction and package top surface, and is estimated from JEDEC thermal tests [11]. There is, however, uncertainty in T_T due to various noise sources. We overcome this problem by modeling T_T readings as a stochastic process.

Power dissipation of logic devices in the substrate is the major source of heat generation. Leakage power dissipation has become an important contributor to the total power dissipation of a VLSI design. Furthermore, process, voltage and temperature (PVT) variations result in statistical changes in the spatial distribution of power dissipation across the die, i.e., the power density and the resulting temperature profile will have different values from one part of the chip to another part and from one time instance to next on the same location in the chip. As a result, power dissipation, which is affected by PVT variation as well as time and space dependent gate-level switching activities in the circuit, cannot be easily characterized by the design itself.

The key contribution of the proposed framework is to recognize the uncertainty in estimating the power state of the system and the resulting junction temperature of the IC, and to qualitatively manage this uncertainty by providing timely alerts about the local hot spots.

C. Temperature Estimation Framework

It is useful to describe how the KF can be adapted to our proposed framework, where our goal is to estimate the junction temperature of a device.

Definition 1: Kalman Filter-based Temperature Estimation (KFTE) framework. The KFTE is tuple $(s, a, o, \mathbf{X}, \mathbf{Y}, \mathbf{Z})$ where

- s is a state representing the junction temperature, T_j ,
- a is a voltage and frequency assignment (VFA) action,
- o is a temperature observation, T_T ,

- \mathbf{X} denotes a state transition matrix,
- \mathbf{Y} denotes an action-input matrix, and
- \mathbf{Z} denotes an observation matrix.

We assume that a power manager (e.g., the operating system) commands an appropriate action a , which changes the operational mode (i.e., power state) of the design, and hence, will result in a change in its T_j value (i.e., a change in state s). In our proposed framework, we estimate the next value of T_j (state s') by employing a prediction technique based on the KF algorithm, while analyzing (possibly noisy) T_T values. The state and observation calculations are performed by using the following linear matrix equations:

$$s^{t+1} = \mathbf{X} s^t + \mathbf{Y} a^t + u^t, \quad u^t \sim N(0, Q^t) \quad (6a)$$

$$o^{t+1} = \mathbf{Z} s^{t+1} + v^{t+1}, \quad v^{t+1} \sim N(0, R^t) \quad (6b)$$

where t denotes a time step, u^t is a temperature state noise which is normally distributed with zero mean and variance Q^t , v^{t+1} is a temperature observation noise normally distributed with zero mean and variance R^t . The state transition matrix \mathbf{X} includes the probabilities of transitioning from state s^t to another state s^{t+1} when action a^t is taken, the action-input matrix \mathbf{Y} relates the action input to the state, whereas the observation matrix \mathbf{Z} , which maps the true state space into the observed space, contains the probabilities of making observation o^{t+1} when action a^t is taken, leading the system to enter state s^{t+1} . In practice \mathbf{X} , \mathbf{Y} and \mathbf{Z} might change with each time step or measurement, but here we assume they are constant. Furthermore, we assume that the initial state, and the noise vectors at each step $\{s^0, u^1, \dots, u^t, v^1, \dots, v^t\}$ are mutually independent.

KFTE tries to obtain an estimate of the junction temperature from the T_T data. The estimation procedure for the temperature state consists of prediction and update phases as is explained in section 4.

D. Power Profile Estimation Framework

We assume that the target system has k power states denoted by pwr_1, \dots, pwr_k , where $pwr_1 < \dots < pwr_k$ in terms of power dissipation values in the respective states. In the context of system modeling under uncertainty, a belief state b is the posterior distribution of the underlying power state given observations and actions. Thus, we use POMDP to formulate a power estimation framework as described below.

Definition 2: POMDP-based Power Profile Estimation (P3E) framework. The P3E is a tuple (b, a, o, T, Z) such that

- b is a belief state about power dissipation level of the system,
- a is an action input, e.g., VFA,
- o is an observation, e.g., temperature value T_T ,
- T is a state transition function, and
- Z is an observation function

A belief state, b^{t+1} , after action a^t and observation o^{t+1} , may be calculated from the previous belief state b^t as follows:

$$b^{t+1} = \frac{Z(o^{t+1}, a^t, pwr^{t+1}) \sum_{pwr} b^t T(pwr^{t+1}, a^t, pwr)}{\sum_{pwr, pwr'} Z(o^{t+1}, a^t, pwr') b^t T(pwr', a^t, pwr)} \quad (7)$$

Note that the ranges of temperature measurement by a thermal sensor are defined by the temperature thresholds from ACPI (Advanced Configuration and Power Interface) specification.

The estimation of power state of the system is performed by

obtaining the maximum a posteriori (MAP) estimate based on the Bayesian approach, which provides a way to include the prior knowledge concerning the quantities to be estimated, as will be explained in section 4.

Figure 3 illustrates the proposed uncertainty-aware estimation framework, where the estimators are based on the KF algorithm for the junction temperature of the device and based on the POMDP for the power state of the system. Assume that the operating system, based on performance requirements, can choose an action from a finite set of action $A = \{a_1, \dots, a_n\}$, where $a_1 < \dots < a_n$ in terms of voltage and frequency values. These actions are taken at periodic time instances (synchronous) or interrupt-based event occurrences (asynchronous), which are called *decision epochs*.

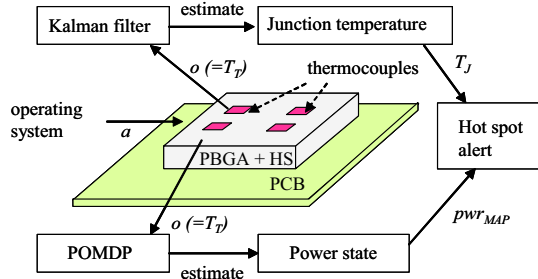


Figure 3. Uncertainty-aware estimation framework.

IV. Hot Spot Alerting Algorithm

In this section, we first explain the method for estimating the junction temperature of the chip as well as the power state of the system in an uncertain environment (cf. Figure 4).

We point out that the Kalman filter is a recursive estimator, which means that only the estimated state from the previous time step and the current measurement are needed to generate the estimate for the current state. The Kalman filter has two distinct phases: Predict and Update. The predict phase uses the state estimate from the previous time step to produce an (a priori) estimate of the state at the current time step. In the update phase, measurement information at the current time step is used to refine this prediction to arrive at a new, (hopefully) more accurate (a posteriori) state estimate, again for the current time step.

A. Estimation of junction temperature of the chip

In KFTE, the framework performs the temperature estimation based on the KF as follows.

- Initialize*: The algorithm initializes the first state s^t to s^0 , and the error covariance matrix which is a measure of the estimated accuracy of the state prediction E^t to a diagonal matrix where the diagonal elements are set to some fixed value, signifying that the initial system state is uncertain.
- Predict*: The algorithm computes the predicted (a priori) state s_-^{t+1} and the predicted (a priori) error covariance matrix E_-^{t+1} .
- Update*: The algorithm first computes the optimal Kalman gain K^{t+1} and uses it to produce an updated (a posteriori) state estimate, s^{t+1} , as a linear combination of s_-^{t+1} and the Kalman gain-weighted residue between an actual

observation o^{t+1} and the predicted observation $Z s_-^{t+1}$. The algorithm also updates the error covariance matrix. Please refer to [10] for details of prediction and correction stages.

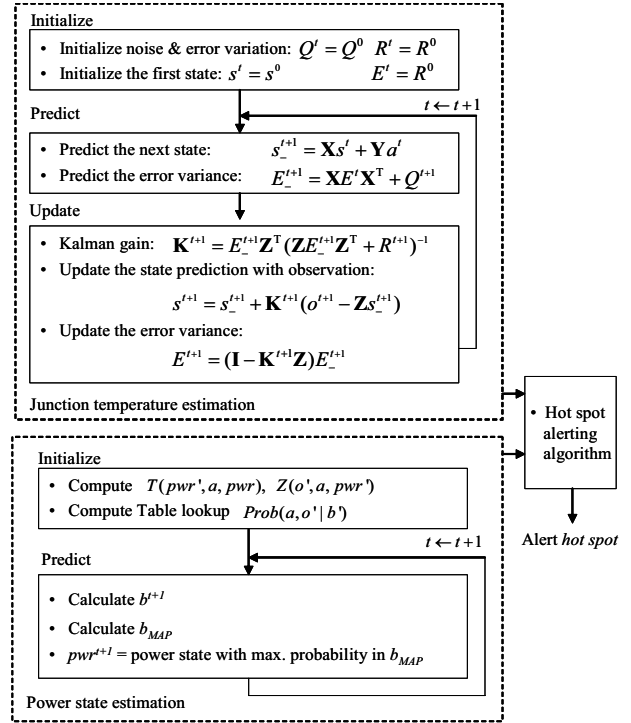


Figure 4. The flow of the proposed estimation technique.

B. Estimation of power state of the system

We consider only the task of estimating the system power state, not controlling it, where the system generates temperature observation $o \in O$ given an action a . Let history, h , denote a stream of action-observation pairs which characterize the system behavior as $h^{0:t} := (a^0, o^1, a^1, o^2, \dots, a^{t-1}, o^t)$. Then, according to the Bayesian formula, the probability density function of belief state b given h can be written as

$$Prob(b^t | h^t) = \frac{Prob(h^t | b^t) \cdot Prob(b^t)}{Prob(h^t)} \quad (8)$$

where $Prob(b^t | h)$ is called the posterior probability density function (PPDF), $Prob(h^t | b^{t-1})$ is the likelihood function, $Prob(h^t)$ is the prior distribution, and $Prob(b^t)$ is precisely the probability of belief state which can be obtained from Eqn. (7). Once the PPDF is known, the most probable power state can be computed as follows

$$\begin{aligned} b_{MAP} &= \arg \max_b Prob(b^t | h^t) \\ &= \arg \max_b Prob(h^t | b^t) \cdot Prob(b^t) \end{aligned} \quad (9)$$

Note that as a normalizing constant, the knowledge of $Prob(h^t)$ is not needed because we are not interested in making any decisions. Since we assume that action a is issued to the system at each decision epoch, we may consider that the current power state of the system is only affected by the previous action and observation, which results in

$$b_{MAP} = \arg \max_b Prob(a^{t-1}, o^t | b^t) \cdot Prob(b^t) \quad (10)$$

where we use a table-lookup method for obtaining $Prob(a^{t-1}, o^t | b^t)$ efficiently. Note that even though we know the given action a^{t-1} with certainty, the observation o^t is only known probabilistically. Let pwr_{MAP} denote the power-state which has the maximum probability in the belief state, b_{MAP} , obtained from Eqn. (10).

C. Hot spot alerting algorithm

In predicting hot spots, we combine estimation for the junction temperature of the chip (by using KF) with estimation for power dissipation of the system (by using POMDP).

We assume the presence of a thermal sensor which produces a stream of continuous-valued temperature readings that are noisy. The thermal time constant of the sensor, i.e., the time it takes for the thermal sensor to produce a reading of temperature in the sensing region, is much larger than the circuit clock period [16]. This fact in turn implies that recognizing a temperature rise by using a thermal sensor may render the thermal control mechanism useless due to its slow response. Therefore, we propose a hot spot alerting algorithm based on the predictions of the junction temperature of the device and the power state of the system.

```

1: do forever
2:   predict the junction temperature,  $T_j^{t+1}$ 
3:   predict the power state of the processor,  $pwr^{t+1}$ 
4:   if  $T_j^{t+1} \geq T_{aH}$ 
5:     alert red hot spot
6:   else if  $T_{aL} \leq T_j^{t+1} < T_{aH}$ 
7:     if  $pwr^{t+1} \geq P_a$ 
8:       alert red hot spot
9:     else
10:      alert yellow hot spot
11:   else
12:     if  $\partial T_j / \partial t \geq G_{ja}$ 
13:       alert yellow hot spot
14: return hot spot level

```

Figure 5. The proposed hot spot alerting algorithm.

Figure 5 shows the proposed hot spot alerting algorithm. We define *red* and *yellow* hot spot levels in terms of the degree of thermal threat. Note that T_{aH} and T_{aL} are pre-defined temperature threshold values ($T_{aL} < T_{aH}$), P_a is the power threshold value, and G_{ja} is the temperature gradient threshold value. All these parameters are set by system or package developers.

V. Experimental Results

The proposed technique is applied to a MIPS-compatible RISC processor with a 5-stage pipeline, instruction / data caches, and internal SRAM for code/data storage. To precisely evaluate the characteristics of the processor, we relied on the detailed Verilog RT-level description of the processor synthesized with a TSMC 65nm cell library. The power dissipation numbers were obtained through functional simulations with exact switching activity information.

We do not have a packaged IC equipped with a thermal sensor to report T_T . Hence, we estimate T_T by combining Eqn. (4) and (5), resulting in $T_T = T_A + P \cdot (\theta_{JA} - \psi_{JT})$. Assuming that $T_A = 70^\circ\text{C}$ and using package thermal performance data of Table 3 for θ_{JA} and ψ_{JT} . Note that device power P in the above equation is assumed to be a normally distributed random

variable with a mean value of P_{sim} and a standard deviation of ΔP . Now, P_{sim} is the simulated power number while ΔP is the standard deviation of power values, which is calculated by running different tasks on the processor at different process corners (e.g., fast, typical, and slow) available with the TSMC 65nm library. We thus generate different T_T values by running various benchmark programs, regularly monitoring and recording P_{sim} values, but subsequently using a power value P which follows a normal distribution, $N(P_{sim}, (\Delta P)^2)$.

We first analyzed the power consumption inside the processor by executing SPECint2000 benchmarks [17] as reported in Table 1 (without accounting for memory power). This table indicates that non-uniform power density exists across the processor, which impacts local hot spot on a die.

Table 1 Percentage of power consumption in different modules of a MIPS-like processor

SPECint2000	Functional modules								
	i-areg	d-areg	incr	mul	alu	sr	reg	decode	busctl
gcc	14.4%	13.8%	4.1%	8.3%	17.6%	2.2%	16.4%	8.1%	15.1%
gap	13.1%	15.7%	4.0%	6.2%	16.2%	1.7%	18.1%	8.4%	16.6%
gzip	9.2%	15.4%	4.6%	7.5%	21.6%	2.3%	17.1%	9.2%	13.1%

Table 2 Definition of power and temperature states of the processor

	power [W] state			observation [$^\circ\text{C}$] state		
	pow_1	pow_2	pow_3	o_1	o_2	o_3
range	[0.6 1.4]	(1.4 2.2]	(2.2 3.0]	[86 93]	(93 100]	(100 107]

Next we set the power (W) and temperature ($^\circ\text{C}$) ranges corresponding to three power states and three temperature states (see Table 2). These values were obtained during the active state of the processor (recall that thermal control occurs mostly during active state) with the extracted thermal data [18] for (31mm x 31mm) PBGA as summarized in Table 3.

Table 3 PBGA 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

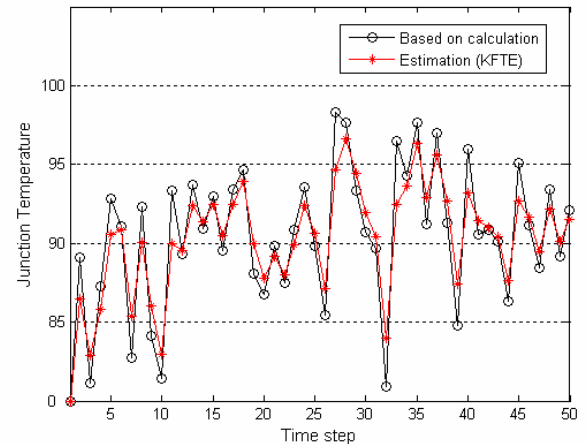


Figure 6. Trace of estimation for the junction temperature.

Figure 6 shows the trace of junction temperature estimation, where we randomly chose a sequence of 50 programs of SPECint2000, which include *gcc*, *gap*, and *gzip*. For example, one such sequence of programs may be *gap₁-gcc₂-gcc₃-...-gzip₄₉-gap₅₀*, where *program_i* is the *i*th program in the sequence, which is executed on the processor. In this simulation setup, for simplicity but without loss of generality, we set that the values of Q' and R' to 1. The time steps are abstractly defined. Note that the temperature trace of calculation in the figure is based on above-mentioned method (i.e., by using the estimate of T_j).

Considering the estimation of power state, Figure 7 shows the trace of belief state for power states, *pow₁*, *pow₂*, and *pow₃*, as estimated by the proposed POMDP-based technique. Here *belief state (pow_i)* denotes the probability we are in the *pow_i* state. For this example data, the power dissipation of the system increases as time advances.

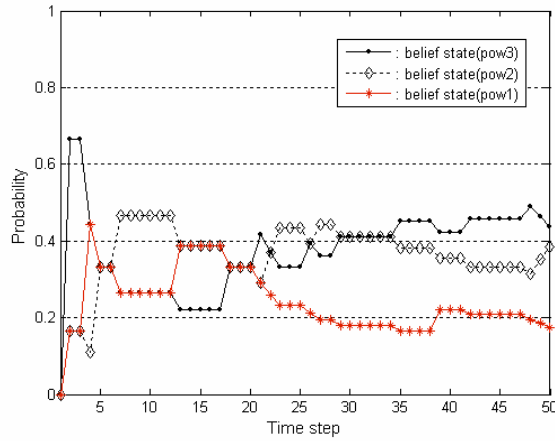


Figure 7. Trace of belief state for the power profile.

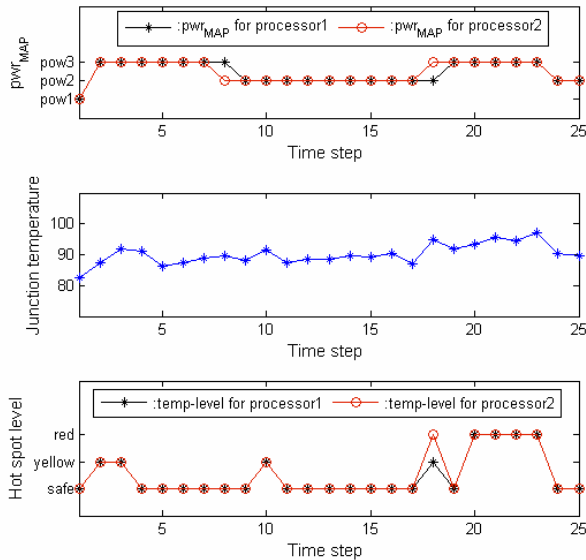


Figure 8. Evaluation of the hot spot alerting algorithm.

Experimental results of the proposed hot spot alerting algorithm are shown in Figure 8. Here we have assumed that two processor core inside a multicore processor execute 50 programs alternately. Hot spot levels are defined as *red alert*, *yellow alert* (cf. Figure 5) and *safe* (i.e., there is no thermal

threat), where we set the required parameter values as: $T_{a,H} = 100^\circ\text{C}$, $T_{a,L} = 90^\circ\text{C}$, $G_{j,a} = 7^\circ\text{C}$, and $P_a = 2.2\text{W}$. The power state of each processor is also estimated as explained above. Results in this figure demonstrate that local hot spots (see Figure 8 bottom) are estimated based on the power state of the processor (see Figure 8 top), considering the junction temperature of the device (see Figure 8 middle).

VI. Conclusions

We have proposed a stochastic hot spot alerting technique based on estimations of the junction temperature of a device and the power state of a system. The proposed uncertainty-aware estimation framework efficiently captures the uncertain dynamics of the system behaviors. Being able to handle various sources of uncertainty would improve the accuracy and robustness of the estimation technique, ensuring the thermal safety of the device with truly quality and reliability. Experimental results demonstrate that the proposed technique alerts thermal threats under probabilistic variations.

REFERENCES

- [1] M. Pedram, and S. Nazarian, "Thermal Modeling, Analysis and Management in VLSI Circuits: Principles and Methods," *Proc. of IEEE*, Vol. 95, No. 8, Aug., 2006.
- [2] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact Thermal Modeling for Temperature-Aware Design," *Proc. of Design Automation Conference*, Jun., 2004.
- [3] D. Brooks, and M. Martonosi, "Dynamic Thermal Management for High Performance Microprocessor," *Proc. of High Performance Computer Architecture*, Jan., 2001.
- [4] J. Srinivasan, and S. V. Adve, "Predictive Dynamic Thermal Management for Multimedia Applications," *Proc. of 17th ACM Int'l Conference on Supercomputing*, Jun., 2003.
- [5] A. Naveh, et al., "Power and Thermal Management in Intel Core Duo Processor," *Intel Technology Journal*, Vol. 10, Issue 2, May, 2006.
- [6] Dual-Core Processor Power and Thermal Design Guide, Mar., 2006. <http://www.amd.com>.
- [7] P. Dadvar, and K. Skadron, "Potential Thermal Security Risks," *Proc. of 21st IEEE Semi-Therm Symposium*, Mar., 2005.
- [8] M. Janicki, and A. Napieralski, "Inverse Heat Conduction Problems in Electronics with Special Considerations of Analytical Analysis Methods," *Proc. of Int'l Semiconductor*, Vol. 2, Issue 4, Oct., 2004.
- [9] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. Wiley Publisher, New York, 1994.
- [10] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problem," *Journal of Basic Engineering*, Vol. 82, Series D, 1960.
- [11] H. Shaukatullah, and A. Claassen, "Effect of Thermocouple Wire Size and Attachment Method on Measurement of Thermal Characteristics of Electronic Package," *Proc. of 19th IEEE Semi-Therm Sym*. 2003.
- [12] JEDEC standards. <http://www.jedec.org>.
- [13] E. Rotem, J. Hermerding, C. Aviad, and C. Harel, "Temperature measurement in the Intel Core Duo Processor," *Proc. of 12th Int'l Workshop on Thermal investigation of ICs*, Sep., 2006.
- [14] Y. Cheng, C. Tsai, C. Teng, and S. Kang, *Electrothermal Analysis of VLSI Systems*, Kluwer Academic Publishers, 2000.
- [15] K. Skadron, et al., "Temperature-Aware Microarchitecture," *Proc. of Int'l Symposium on Computer Architecture*, Jun. 2003.
- [16] T. Wang, Y. Lee, and C. C. Chen, "3D Thermal-ADI: An Efficient Chip-Level Transient Thermal Simulator," *Proc. of ISPD*, Apr., 2003.
- [17] CPU SPECint2000 documents. <http://www.spec.org>.
- [18] Thermal data for MIPS64 processors. <http://www.broadcom.com>.