A Stochastic Local Hot Spot Alerting Technique

Hwisung Jung and Massoud Pedram

University of Southern California
Dept. of Electrical Engineering
Overview

- Motivation and Background
- Uncertainty-Aware Estimation Framework
- The Proposed Hot Spot Alerting Algorithm
- Experimental Results
- Conclusion
Introduction

- As IC process geometries shrink below 65nm
  - Higher power density
  - Higher operating temperature
  - Lower circuit reliability

- Thermal management is a major design requirement
  - Elevated temperature increases leakage and lowers performance
  - Gate oxide lifetime is highly dependent on the $T_J$ of IC

- Local hot spots are becoming more prevalent
  - Thermal runaway; CMOS latch-up
  - Functional and/or timing errors; accelerated aging

Timely identification and elimination of hot spots is a key goal
Some Relevant Prior Work

- K. Skadron, et al. (ISCA 2003)
  - An architectural-level thermal model, *HotSpot*

- W. Huang, et al. (DAC 2004)
  - A compact thermal model for temperature-aware design

- D. Brook, et al. (HPCA 2001)
  - A thermal control mechanism, *Wattch*

- J. Srinivasan, et al. (ICS 2003)
  - Predictive dynamic thermal management

- R. Mukherjee, et al. (DAC 2006)
  - Thermal sensor allocation and placement
Many researchers have examined techniques for:

- Thermal modeling considering the chip-package-heat sink interface characteristics
- Dynamic thermal management to control rapid temperature rise

These techniques suffer from the following:

- Thermal modeling based on equivalent circuit models cannot accurately account for real structures with complex shapes and boundary conditions
- Thermal management utilizing external thermal sensors tend to provide readings which are often late and/or are inaccurate
- On-chip sensors are expensive to employ and suffer from noise, nonlinearity, and low response speed

There is thus uncertainty in identifying local hot spots

- Stochastic modeling and early prediction of local (zone-based) temperatures is a necessity
IC package can be characterized by its thermal resistance

- Heat is transferred from the die into the ambient air
- Value of the thermal resistance along with the on-chip power dissipation determines the temperature rise of the junction above a reference point

Heat flow in the PBGA + HS package

One of the IC package heat transfer paths and the corresponding thermal resistive model
Thermal resistance is defined as

\[ \theta_{jX} = \frac{(T_J - T_X)}{P} \]

- \( \theta_{jX} \) is the thermal resistance from device junction to specific point
- \( T_J \) is the device junction temperature
- \( T_X \) is the temperature of some specific reference point
- \( P \) is the device power dissipation

When the reference points are selected as the ambient air, PCB, and case top, we have

\[ \begin{align*}
\theta_{JA} &= \frac{(T_J - T_A)}{P} \\
\theta_{JB} &= \frac{(T_J - T_B)}{P} \\
\theta_{JC} &= \frac{(T_J - T_C)}{P}
\end{align*} \]

- Junction-to-air
- Junction-to-board
- Junction-to-case

- \( T_A, T_B \) and \( T_C \) denote the temperatures of ambient air, PCB, and case top
Junction-to-air thermal resistance may be calculated as

\[ \theta_{JA} = \left( \frac{1}{\theta_{JB}} + \frac{1}{\theta_{BS}} + \frac{1}{\theta_{BA}} \right)^{-1} \]

- Junction temperature is estimated as:
  \[ T_J = T_A + P \cdot \theta_{JA} \]
- The goal of thermal design is to make the \( \theta_{JA} \) value small so that the junction temperature \( T_J \) does not exceed some specified maximum value.

\( \theta_{JA} \) cannot be modeled directly due to the complexity of thermal models for the substrate, package, and board stack and the cooling system which is in-place.

- In addition, \( \theta_{JA} \) is typically taken to be a single parameter under the assumption that \( P \) is distributed uniformly across the die, which is unrealistic.
Overview of the Proposed Solution

- We develop a hot spot alerting technique by estimating the *junction temperature* and the *system power state*
  - The thermal time constant of the die is larger than the circuit clock speed
  - Recognizing a temperature rise by relying on thermal sensors and subsequently employing thermal control mechanisms can result in too late a response (i.e., a corrective action to avoid thermal problems)

- Our proposed hot spot alerting technique combines:
  - State estimation for the junction temperature using Kalman Filter (KF)
  - State estimation for the system power dissipation using Partially Observable Markov Decision Process (POMDP) model
Use (external temperature) observations to estimate the \textit{Junction Temperature} ($T_J$) and the \textit{Power state} ($pwr_{MAP}$). 

Uncertainty-aware hot spot warning framework
Temperature Estimation

- **Kalman Filter**
  - Estimate the state of a system based on the previous state, previous action, and the current observation

- **Kalman Filter-based Temperature Estimation (KFTE) Framework**
  - $s$ is a state representing the junction temperature $T_J$
  - $a$ is an action (voltage-frequency assignment) issued by the DP/TM
  - $o$ is a temperature observation $T_T$
  - $X$ denotes a state transition matrix
  - $Y$ denotes an action-input matrix
  - $Z$ denotes an observation matrix

\[
s^{t+1} = Xs^t + Ya^t + u^t, \quad u^t \sim N(0, Q^t) \quad u: \text{the temperature variation}
\]

\[
o^{t+1} = Zs^{t+1} + v^{t+1}, \quad v^{t+1} \sim N(0, R^t) \quad v: \text{the observation noise}
\]
Estimation of the junction temperature of the chip

- Based on the Kalman Filter

Initialize

- Initialize noise & error variation: $Q^i = Q^0$, $R^i = R^0$, $E^i = R^0$
- Initialize the first state: $s^i = s^0$

Predict

- Predict the next state: $s^i_{t+1} = Xs^i_t + Ya^t$
- Predict the error variance: $E^i_{t+1} = XE^i_t X^T + Q^i_{t+1}$

Update

- Kalman gain: $K^i_{t+1} = E^i_{t+1} Z^T (ZE^i_{t+1} Z^T + R^i_{t+1})^{-1}$
- Update the state prediction with observation:
  $s^i_{t+1} = s^i_{t+1} + K^i_{t+1} (a^i_{t+1} - Zs^i_{t+1})$
- Update the error variance: $E^i_{t+1} = (I - K^i_{t+1} Z)E^i_{t+1}$

Junction temperature estimation
Power State Estimation

- POMDP (Partially Observable Markov Decision Process)
  - It is a model for deciding how to act in an accessible, stochastic environment with a known transition model
  - It can model uncertainty and non-determinism in observations

- POMDP is a tuple $<S, A, O, T, Z>$ such that
  - $S$ is a finite set of discrete states
  - $A$ is a finite set of actions
  - $O$ is a finite set of observations
  - $T$ is a state transition probability function
  - $Z$ is an observation function

- POMDP may be transformed to a regular (continuous state) MDP by defining and maintaining a belief state, $b^t$
  - Vector $b$ gives the probability distribution over the possible states of the system, $\sum_{s \in S} b^t(s) = 1$
Estimation of the power state of the system

- Based on the Bayesian formula:

\[
Prob(b^t \mid h^t) = \frac{Prob(h^t \mid b^t) \cdot Prob(b^t)}{Prob(h^t)}
\]

- \( h \) is a stream of action-observation pairs
- \( Prob(b^t \mid h) \) is the posterior probability density function
- \( Prob(h^t \mid b^t) \) is the likelihood function
- \( Prob(h^t) \) is the prior distribution
- \( Prob(b^t) \) is the probability of belief state

The most probable power state can be computed as the maximum a posterior (MAP) estimate under Markovian assumption as:

\[
b_{MAP} = \arg \max_{b \in B} Prob(b^t \mid h^t) = \arg \max_{b \in B} Prob(h^t \mid b^t) \cdot Prob(b^t) = \arg \max_{b \in B} Prob(a^{t-1}, o^t \mid b^t) \cdot Prob(b^t)
\]
Flow of the Estimation Framework

**Initialize**
- Initialize noise & error variation: \( Q^0 = Q^\prime \), \( R^0 = R^\prime \)
- Initialize the first state: \( s^0 = s^\prime \)

**Predict**
- Predict the next state: \( s_{t+1}^0 = Xs^t + Ya^t \)
- Predict the error variance: \( E_{t+1}^0 = XE^tX^T + Q^t \)

**Update**
- Kalman gain: \( K_{t+1} = E_{t+1}^0Z^T(ZE_{t+1}^0Z^T + R_{t+1}^0)^{-1} \)
- Update the state prediction with observation:
  \[ s_{t+1} = s_{t+1}^0 + K_{t+1}(o^t - Zs_{t+1}^0) \]
- Update the error variance:
  \[ E_{t+1} = (I - K_{t+1}Z)E_{t+1}^0 \]

**Power state estimation**
- Inform about the hot spot level

**Junction temperature estimation**
The proposed hot spot alerting algorithm

- Define red and yellow hot spot levels in terms of degree of thermal threat
- \( T_{a,H} \) and \( T_{a,L} \) are pre-defined temperature thresholds (\( T_{a,L} < T_{a,H} \))
- \( P_a \) is a power dissipation threshold
- \( G_{j,a} \) is a temp. gradient threshold

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 \( T_{j,t+1} \geq T_{a,H} \)
5: alert red hot spot
6: else if \( T_{a,L} \leq T_{j,t+1} < T_{a,H} \)
7: if \( pwr_{t+1} \geq P_a \)
8: alert red hot spot
9: else
10: alert yellow hot spot
11: else
12: if \( \partial T_j / \partial t \geq G_{j,a} \)
13: alert yellow hot spot
14: return hot spot level
Experimental Setup

- The technique is applied to a 32-bit RISC processor
- Set the parameter values for estimation framework

<table>
<thead>
<tr>
<th></th>
<th>power [W] state</th>
<th>observation [°C] state</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p_{ow_1}$</td>
<td>$p_{ow_2}$</td>
<td>$p_{ow_3}$</td>
</tr>
<tr>
<td>range</td>
<td>[0.6 1.4]</td>
<td>(1.4 2.2]</td>
</tr>
<tr>
<td></td>
<td>(2.2 3.0]</td>
<td>[86 93]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(93 100]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(100 107]</td>
</tr>
</tbody>
</table>

- PBGA package thermal performance data ($T_A$=70°C)

<table>
<thead>
<tr>
<th>Air velocity</th>
<th>$T_{J_{max}}$ [°C]</th>
<th>$T_{T_{max}}$ [°C]</th>
<th>$\psi_{JT}$ [°C/W]</th>
<th>$\theta_{JA}$ [°C/W]</th>
</tr>
</thead>
<tbody>
<tr>
<td>m/s ft/min</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.51 100</td>
<td>107.9</td>
<td>106.7</td>
<td>0.51</td>
<td>16.12</td>
</tr>
<tr>
<td>1.02 200</td>
<td>105.3</td>
<td>104.1</td>
<td>0.53</td>
<td>15.62</td>
</tr>
<tr>
<td>2.03 300</td>
<td>102.7</td>
<td>101.2</td>
<td>0.65</td>
<td>14.21</td>
</tr>
</tbody>
</table>

[$\psi_{JT}$: Junction-to-top of package thermal characterization parameter]
Choose a sequence of 50 application programs

- E.g., gap\textsubscript{1} - gzip\textsubscript{2} - gap\textsubscript{3} - gcc\textsubscript{4} - ... - gap\textsubscript{50}.

Trace of the junction temperature

- Golden result:
  - Calculate $T_T$ from $T_T = T_A + P \cdot (\theta_{JA} - \psi_{JT})$, where $P \sim N(P_{sim}, (\Delta P)^2)$

- Estimated result:
  - Based on KFTE
Experimental Results (2/3)

- Trace of the power belief state
  - e.g., belief state($pow_1$): probability for power state $pow_1$
  - Evaluated by the POMDP-based Power Profile Estimation (P3E) method
Evaluation of the proposed hot spot alerting algorithm

- Hot spot levels defined: red, yellow, and safe

![Graphs showing hot spot levels and temperature trends over time.](image-url)
Conclusion

- The stochastic hot spot alerting technique based on
  - estimation of the junction temperature of the device and
  - prediction of the power state of the system

- The proposed uncertainty-aware estimation framework efficiently captures
  - stochastic behavior of the system
  - PVT variations in system performance parameters, and
  - inaccuracies in temperature measurements

- Experimental results demonstrate that the proposed technique provides early warning about thermal threats under large variations