

An Information-theoretic Framework for Optimal Temperature Sensor Allocation and Full-chip Thermal Monitoring

Huapeng Zhou and Xin Li
Carnegie Mellon University
Pittsburgh, PA 15213
{huapeng, xinli}@cmu.edu

Chen-Yong Cher, Eren Kursun and Haifeng Qian
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
{chenyong, ekursun, qianhaifeng}@us.ibm.com

Shi-Chune Yao
Carnegie Mellon University
Pittsburgh, PA 15213
sy0d@andrew.cmu.edu

ABSTRACT

Full-chip thermal monitoring is an important and challenging issue in today's microprocessor design. In this paper, we propose a new information-theoretic framework to quantitatively model the uncertainty of on-chip temperature variation by differential entropy. Based on this framework, an efficient optimization scheme is developed to find the optimal spatial locations for temperature sensors such that the full-chip thermal map can be accurately captured with a minimum number of on-chip sensors. In addition, several efficient numerical algorithms are proposed to minimize the computational cost of the proposed entropy calculation and optimization. As will be demonstrated by our experimental examples, the proposed entropy-based method achieves superior accuracy (1.4× error reduction) for full-chip thermal monitoring over prior art.

Categories and Subject Descriptors

B.7.2 [Integrated Circuits]: Design Aids – Verification

General Terms

Algorithms

Keywords

Thermal Monitoring, Integrated Circuit

1. INTRODUCTION

As integrated circuit (IC) technology continues to scale to the nanoscale era, power and thermal issues become increasingly important for microprocessor design. In general, modern microprocessors can be classified into two broad categories [1]: (a) power-limited microprocessors, and (b) hotspot-limited microprocessors. The operation of power-limited microprocessors is constrained by the availability of power supply such as the limited battery life and/or the capacity of the power delivery network. Most embedded microprocessors fall into this category. On the other hand, the performance of hotspot-limited microprocessors, such as the high-performance microprocessors used for enterprise servers, is constrained by their cooling capability. The high temperature at hotspots directly impacts the device lifetime and the circuit reliability [2].

During the past decade, a large number of thermal management approaches have been proposed [2]-[8] to apply circuit-level and/or architecture-level techniques (e.g., power gating, dynamic voltage and frequency scaling, clock throttling, etc.) to optimally explore the trade-offs between microprocessor performance and hotspot temperature. These techniques have been

successfully integrated into many commercial microprocessors and thermal management has become a critical component for advanced microprocessor design.

The efficacy of thermal management, however, heavily relies on the accuracy of on-chip temperature estimation and prediction. Due to localized heating, the temperature variation within a single chip can reach up to 10s of degrees. For instance, within-die temperature variation up to 50 °C was reported in [9]. In order to accurately capture the within-die temperature variation for the emerging processors that become increasingly complex, a large number of temperature sensors must be deployed throughout the chip to collect thermal data in real time. These temperature sensors along with their peripheral circuits introduce substantial overhead in silicon area, power consumption and design complexity. Hence, it is extremely important to minimize the design overhead associated with temperature sensors without surrendering the accuracy of thermal monitoring. In other words, the open question here is how to accurately capture the full-chip thermal map with a *minimum* number of temperature sensors.

To answer this question, two different issues must be carefully addressed. First, an efficient methodology must be developed to identify the optimal spatial locations to deploy a minimum number of on-chip temperature sensors. This goal can be achieved by modeling and analyzing the thermal structure of the microprocessor design, the workload statistics and/or the historical thermal measurement data. Second, once the temperature data are measured by on-chip sensors, an efficient algorithm should be developed to estimate the full-chip thermal map. In the literature, various works have been proposed to address these two issues [10]-[17]. These techniques have been successfully applied to a broad range of practical applications.

In this paper, we propose an information-theoretic framework to address the thermal monitoring problem. Our goal is to fundamentally re-think the on-chip temperature variation from the information point of view and, consequently, develop an optimal solution to solve the thermal monitoring problem. To this end, we propose to accurately model and then quantitatively measure the uncertainty of on-chip temperature variation by differential entropy. Based on this information-theoretic framework, an optimization algorithm is developed to select a set of optimal sensor locations by minimizing the uncertainty (i.e., the differential entropy) of on-chip temperature variation. In other words, once the temperature is measured by on-chip sensors at these selected locations, the full-chip thermal map can be accurately estimated by the proposed information-theoretic framework. As will be demonstrated by the experimental examples in Section 4, the proposed entropy-based approach achieves superior accuracy (around 1.4× error reduction) for full-chip thermal monitoring over other traditional methods.

Another contribution of this paper is to develop an efficient numerical algorithm to reduce the computational complexity for the proposed entropy calculation and minimization. In particular, the proposed numerical algorithm is facilitated by two core

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2012, June 3-7, 2012, San Francisco, California, USA.
Copyright 2012 ACM 978-1-4503-1199-1/12/06...\$10.00

techniques: (a) Schur complement, and (b) incremental Cholesky decomposition. The combination of these two techniques results in $69\times$ runtime speedup, compared to a simple implementation without using these fast algorithms.

The remainder of this paper is organized as follows. In Section 2, we first develop the proposed information-theoretic framework for optimal sensor allocation and thermal map estimation. Next, several numerical algorithms are proposed in Section 3 to facilitate an efficient implementation with low computational cost. The efficacy of the proposed method is demonstrated by both simulation results and industrial measurement data in Section 4. Finally, we conclude in Section 5.

2. INFORMATION FRAMEWORK

2.1 Optimal Sensor Allocation

The proposed thermal monitoring is motivated by the assumption that on-chip temperature variation is spatially correlated. Such spatial correlation exists, because the active and leakage power generated at one spatial location can affect the temperature at multiple locations [2]. Due to the spatial correlation, it is possible to measure the temperature at a few locations and then predict the temperature at other locations.

Mathematically, the spatial temperature variation of a given chip can be modeled by a set of random variables:

$$X = [x_1 \quad x_2 \quad \cdots \quad x_N]^T \quad (1)$$

where x_i denotes the temperature at the i th spatial location and N represents the total number of spatial locations of interest. In this paper, we further assume that the random variable X follows a multivariate Normal distribution:

$$pdf(X) = \frac{1}{\sqrt{(2\pi)^N \det(\Sigma_X)}} \cdot \exp\left[-\frac{1}{2}(X - \mu_X)^T \Sigma_X^{-1} (X - \mu_X)\right] \quad (2)$$

where $pdf(\bullet)$ denotes the probability density function (PDF) of a random variable, $\det(\bullet)$ represents the determinant of a matrix, and $\mu_X \in R^N$ and $\Sigma_X \in R^{N \times N}$ are the mean value and the covariance matrix of the multivariate random variable X respectively. A similar assumption of Normal distribution has been used by several previous works such as [12] and [17]. In practice, the mean μ_X and the covariance Σ_X of the Normal distribution can be estimated by a set of simulated and/or measured thermal maps, as will be illustrated by our experimental examples in Section 4.

Based on the information theory, the uncertainty posed by the multivariate random variable X can be quantitatively measured by its differential entropy [20]:

$$H(X) = -\int pdf(X) \cdot \log[pdf(X)] \cdot dX \quad (3)$$

Since X follows a multivariate Normal distribution, substituting (2) into (3) yields:

$$H(X) = \frac{N}{2} \cdot (1 + \log 2\pi) + \frac{1}{2} \cdot \log[\det(\Sigma_X)] \quad (4)$$

Note that Eq. (4) equals the summation of two different terms. The first term is proportional to N (i.e., the dimensionality of X). If a lot of random variables are required to model the variation (i.e., N is large), the differential entropy is large. The second term in (4) is proportional to the determinant $\det(\Sigma_X)$ that directly measures the variance of the distribution. If the variance is large, the differential entropy (i.e., the uncertainty of the variation) is large. Finally, it is worth mentioning that the differential entropy in (4) is independent of the mean value of the Normal distribution. This observation is consistent with our intuition. Namely, the mean value only determines the ‘‘center’’ of the distribution, but it

does not measure the ‘‘uncertainty’’ of the variation.

In our application of thermal monitoring, the objective is to measure a subset of (say, M) random variables and then estimate the values of other $N - M$ random variables. Such estimation is possible, since these random variables are correlated. Without loss of generality, we re-order the random variables in the vector X :

$$Y = W \cdot X = \begin{bmatrix} X_S \\ X_{\bar{S}} \end{bmatrix} \quad (5)$$

where $X_S \in R^M$ contains the random variables that are measured, $X_{\bar{S}} \in R^{N-M}$ contains the other random variables that should be estimated, and $W \in R^{N \times N}$ is a permutation matrix. After the permutation, the mean value and the covariance matrix of the new multivariate random variable Y are equal to:

$$\mu_Y = W \cdot \mu_X \quad (6)$$

$$\Sigma_Y = W \cdot \Sigma_X \cdot W^T \quad (7)$$

Once the multivariate random variable X_S is measured:

$$X_S = T_S \quad (8)$$

the conditional probability $pdf(X_{\bar{S}} | X_S)$ models the uncertainty of $X_{\bar{S}}$ after X_S is known. As $X_{\bar{S}}$ and X_S are jointly Normal, it can be proven that $pdf(X_{\bar{S}} | X_S)$ is a multivariate Normal distribution [20]. Its mean value and covariance matrix are [20]:

$$\mu_{\bar{S}|S} = \mu_{\bar{S}} + \Sigma_{\bar{S}S} \cdot \Sigma_{SS}^{-1} \cdot (T_S - \mu_S) \quad (9)$$

$$\Sigma_{\bar{S}|S} = \Sigma_{\bar{S}\bar{S}} - \Sigma_{\bar{S}S} \cdot \Sigma_{SS}^{-1} \cdot \Sigma_{S\bar{S}} \quad (10)$$

where $\mu_S \in R^M$ and $\mu_{\bar{S}} \in R^{(N-M)}$ are the elements of μ_Y :

$$\mu_Y = \begin{bmatrix} \mu_S \\ \mu_{\bar{S}} \end{bmatrix} \quad (11)$$

and $\Sigma_{SS} \in R^{M \times M}$, $\Sigma_{\bar{S}\bar{S}} \in R^{(N-M) \times (N-M)}$, $\Sigma_{S\bar{S}} \in R^{M \times (N-M)}$ and $\Sigma_{\bar{S}S} \in R^{(N-M) \times M}$ are the four sub-matrices of Σ_Y :

$$\Sigma_Y = \begin{bmatrix} \Sigma_{SS} & \Sigma_{S\bar{S}} \\ \Sigma_{\bar{S}S} & \Sigma_{\bar{S}\bar{S}} \end{bmatrix} \quad (12)$$

In (9)-(10), only the mean value $\mu_{\bar{S}|S}$ depends on the measurement T_S . The covariance matrix $\Sigma_{\bar{S}|S}$ is independent of T_S . This is an important property. It implies that the uncertainty of $pdf(X_{\bar{S}} | X_S)$ is *not* dependent on the measured temperature value. Similar to (4), we can use differential entropy to quantitatively measure the uncertainty of $X_{\bar{S}}$ once X_S is known:

$$H(X_{\bar{S}} | X_S) = \frac{N-M}{2} \cdot (1 + \log 2\pi) + \frac{1}{2} \cdot \log[\det(\Sigma_{\bar{S}|S})] \quad (13)$$

Based on the covariance matrix in (10), the block partition in (12) and the theory of Schur complement [21], we have:

$$\det(\Sigma_{\bar{S}|S}) = \frac{\det(\Sigma_Y)}{\det(\Sigma_{SS})} \quad (14)$$

Substituting (7) into (14) yields:

$$\det(\Sigma_{\bar{S}|S}) = \frac{\det(W \cdot \Sigma_X \cdot W^T)}{\det(\Sigma_{SS})} = \frac{\det(W) \cdot \det(\Sigma_X) \cdot \det(W^T)}{\det(\Sigma_{SS})} \quad (15)$$

Since W is a permutation matrix and $\det(W) = \det(W^T) = 1$, Eq. (15) can be further simplified as:

$$\det(\Sigma_{\bar{S}|S}) = \frac{\det(\Sigma_X)}{\det(\Sigma_{SS})} \quad (16)$$

Combining (13) and (16), we have:

$$H(X_{\bar{S}} | X_S) = \frac{N-M}{2} \cdot (1 + \log 2\pi) + \frac{1}{2} \cdot \log\left[\frac{\det(\Sigma_X)}{\det(\Sigma_{SS})}\right] \quad (17)$$

Based on the differential entropy in (17), our optimal sensor

allocation problem can be mathematically stated as follows: *Given a set of correlated random variables $X = [x_1, x_2, \dots, x_N]^T$, we want to identify and then measure an optimal subset of (i.e., M) random variables in X_S such that the differential entropy $H(X_S | X_S)$ (i.e., the uncertainty posed by the remaining random variables in X_S) is minimized.*

Studying (17), we would have two important observations. First, the differential entropy $H(X_S | X_S)$ is independent of the measured value of X_S . This observation is consistent with the fact that the covariance matrix of the conditional probability $pdf(X_S | X_S)$ is independent of the measurement of X_S , as shown in (10). It, in turn, facilitates us to minimize $H(X_S | X_S)$ without knowing the measured value of X_S . Second, only the denominator of the second term in (17) depends on the random variables that are selected for measurement. Hence, in order to minimize the differential entropy $H(X_S | X_S)$, we should maximize the determinant $det(\Sigma_{SS})$.

Algorithm 1: Optimal Sensor Allocation

1. Start from a set of correlated random variables $X = [x_1, x_2, \dots, x_N]^T$ representing the temperature at different spatial locations, and a given number M representing the total number of temperature sensors.
2. Set $X_S = []$ and $X_S = [x_1, x_2, \dots, x_N]^T$, implying that no random variable is initially selected. Set the iteration index $m = 0$.
3. Find the optimal random variable x_k from the vector X_S such that the determinant of the covariance matrix Σ_{SS} is maximized after adding x_k to the vector X_S .
4. Remove x_k from the vector X_S and add it to the vector X_S .
5. Update the iteration index $m = m + 1$.
6. If $m = M$, stop iteration. Otherwise, go to Step 3.

Optimally finding the M random variables out of the N candidates, however, is a combinatorial optimization problem that is not trivial to solve. In this paper, we propose to apply a greedy algorithm, as shown in Algorithm 1. It selects one random variable to maximize the determinant $det(\Sigma_{SS})$ and, equivalently, minimize the differential entropy $H(X_S | X_S)$ at each iteration step. Such an iterative process continues until M random variables are selected (i.e., the spatial locations of M temperature sensors are determined). The proposed greedy heuristic does not necessarily converge to the global optimum. However, as will be demonstrated by the experimental examples in Section 4, the proposed greedy algorithm reliably finds a set of good sensor locations and it substantially outperforms other traditional algorithms for sensor allocation.

Note that Algorithm 1 requires the total number of temperature sensors (i.e., M) as its input. In practice, the appropriate value of M should be determined by exploring the trade-offs between the accuracy of thermal monitoring and the overhead of circuit design. In other words, if a lot of temperature sensors are used (i.e., M is large), we can achieve high accuracy of thermal map estimation; however, the design overhead (e.g., the silicon area and the power consumption associated with the temperature sensors and their peripheral circuits) is also large. These design trade-offs will be further illustrated by our experimental examples in Section 4.

2.2 Thermal Map Estimation

Once the sensor locations are determined, we can measure the temperature $X_S = T_S$ at these locations. Our next goal is to estimate the value of X_S , i.e., the temperature at other spatial locations that are not measured.

To this end, we need to re-visit the condition probability

$pdf(X_S | X_S)$ that has been discussed in Section 2.1. Remember that $pdf(X_S | X_S)$ models the distribution of X_S after X_S is known. As previously mentioned, $pdf(X_S | X_S)$ is a multivariate Normal distribution. Its mean value and covariance matrix are shown in (9) and (10) respectively. The distribution $pdf(X_S | X_S)$ implies that the value of X_S is not deterministic. In other words, since X_S and X_S are not fully correlated, it is impossible to exactly know X_S after X_S is measured.

While X_S cannot be uniquely determined, it is possible to statistically estimate X_S based on its probability distribution. In this paper, we aim to find the maximum-likelihood estimation (MLE) of X_S , i.e., the value of X_S that is most likely to occur. Since $pdf(X_S | X_S)$ follows a multivariate Normal distribution and it reaches the maximum value at its mean, the MLE of X_S is simply equal to the mean value of $pdf(X_S | X_S)$, as shown in (9). Namely, substituting the measurement $X_S = T_S$ into (9) yields the estimated temperature X_S at other spatial locations.

Finally, it is worth mentioning that a similar statistical method has been proposed in [18] and applied to low-cost characterization of wafer-level process variation. In this paper, we extend the information-theoretic framework to full-chip thermal monitoring. Note that our thermal monitoring problem is substantially different from the variation characterization problem addressed in [18]. For example, due to the high dimensionality of the thermal monitoring problem, it is necessary to further develop efficient numerical algorithms to make the proposed entropy calculation and minimization practically tractable. These implementation details will be discussed in Section 3.

3. IMPLEMENTATION DETAILS

The proposed entropy calculation and minimization are made computationally efficient by implementing a number of fast numerical algorithms. In this section, we describe these implementation issues in detail.

3.1 Schur Complement

Studying Algorithm 1, we notice that Step 3 is the most computationally expensive part of Algorithm 1. During this step, the optimal random variable x_k should be selected from the vector X_S . Towards this goal, the determinant of the following matrix Φ_k must be repeatedly calculated for each random variable x_k that belongs to the vector X_S :

$$\Phi_k = \begin{bmatrix} \Sigma_{SS} & \Sigma_{Sk} \\ \Sigma_{kS} & \Sigma_{kk} \end{bmatrix} \quad (18)$$

where Σ_{SS} is the covariance matrix for all random variables belonging to the vector X_S , Σ_{kS} and Σ_{Sk} are the row and column vectors representing the covariance between the random variable x_k and those random variables in X_S , and Σ_{kk} is a scalar denoting the variance of the random variable x_k . Once the determinant $det(\Phi_k)$ is calculated for different x_k 's, the optimal x_k with the maximum determinant will be selected.

In general, the determinant of the symmetric, positive semi-definite covariance matrix Φ_k can be computed by using Cholesky decomposition [21]. Such a simple implementation can be expensive, because the determinant $det(\Phi_k)$ must be repeatedly calculated for a large number of different x_k 's. To minimize the computational cost, we expand $det(\Phi_k)$ by using the theory of Schur complement [21]:

$$det(\Phi_k) = det(\Sigma_{SS}) \cdot det(\Sigma_{kk} - \Sigma_{kS} \cdot \Sigma_{SS}^{-1} \cdot \Sigma_{Sk}). \quad (19)$$

Studying (19), we would have two important observations.

First, Σ_{SS} , a symmetric and positive semi-definite matrix, is independent of the random variable x_k that is selected. Hence, finding the optimal x_k to maximize the determinant $\det(\Phi_k)$ is equivalent to maximizing the determinant $\det(\Sigma_{kk} - \Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk})$. In other words, only $\det(\Sigma_{kk} - \Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk})$ should be repeatedly calculated for different x_k during the search process.

Second, the matrix inverse Σ_{SS}^{-1} can be pre-computed. Once Σ_{SS}^{-1} is known, the determinant $\det(\Sigma_{kk} - \Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk})$ can be efficiently calculated by the following three numerical operations:

- *Matrix-vector multiplication*: The matrix Σ_{SS}^{-1} is multiplied by the column vector Σ_{Sk} , resulting in a column vector $\Sigma_{SS}^{-1} \Sigma_{Sk}$.
- *Vector-vector multiplication*: The row vector Σ_{kS} is multiplied by the column vector $\Sigma_{SS}^{-1} \Sigma_{Sk}$, resulting in a scalar $\Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk}$.
- *Scalar subtraction*: The scalar $\Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk}$ is subtracted from another scalar Σ_{kk} , resulting in a new scalar $\Sigma_{kk} - \Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk}$. The determinant of a scalar is simply equal to the scalar itself, i.e., $\det(\Sigma_{kk} - \Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk}) = \Sigma_{kk} - \Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk}$.

These numerical operations are substantially cheaper than the traditional Cholesky decomposition, as will be demonstrated by the experimental examples in Section 4.

The key idea of our proposed determinant calculation is to efficiently calculate a large number of determinant values by Schur complement. When implementing the proposed fast algorithm, the matrix inverse Σ_{SS}^{-1} must be efficiently calculated. In what follows, we will further develop an incremental algorithm for Cholesky decomposition that enables us to compute Σ_{SS}^{-1} with low computational cost.

3.2 Incremental Cholesky Decomposition

In this sub-section, we describe the detailed algorithm to calculate the matrix inverse Σ_{SS}^{-1} by using incremental Cholesky decomposition. As shown in Algorithm 1, an optimal random variable x_k is selected and added to the vector X_S at each iteration step. Initially, when X_S only contains one element and Σ_{SS} is a scalar, its inverse Σ_{SS}^{-1} can be easily calculated. Hence, our focus of this sub-section is to develop an efficient numerical algorithm to incrementally update Σ_{SS}^{-1} as more and more random variables are added to the vector X_S over the iterations of Algorithm 1.

Without loss of generality, we consider the case where a random variable x_k is added to the vector X_S at the m th iteration step. After adding x_k to the vector X_S at the end of this iteration step, the matrix Φ_k in (18) is exactly the matrix Σ_{SS} that is required for the next iteration step. Hence, we should calculate the matrix inverse Φ_k^{-1} and assign its value to Σ_{SS}^{-1} .

Our proposed numerical algorithm to calculate Φ_k^{-1} consists of two major steps. First, the matrix Φ_k is incrementally factorized by Cholesky decomposition:

$$\Phi_k = L_\Phi \cdot L_\Phi^T \quad (20)$$

where L_Φ is a lower triangular matrix. During this step, we represent the lower triangular matrix L_Φ by a sub-matrix L_Σ , a row vector $L_{k\Sigma}$ and a scalar L_k :

$$L_\Phi = \begin{bmatrix} L_\Sigma & 0 \\ L_{k\Sigma} & L_k \end{bmatrix}. \quad (21)$$

Substituting (21) into (20) yields:

$$\Phi_k = \begin{bmatrix} L_\Sigma L_\Sigma^T & L_\Sigma L_{k\Sigma}^T \\ L_{k\Sigma} L_\Sigma^T & L_{k\Sigma} L_{k\Sigma}^T + L_k^2 \end{bmatrix}. \quad (22)$$

Comparing (18) and (22), it is easy to verify that the matrix

multiplication $L_\Sigma \cdot L_\Sigma^T$ is simply the Cholesky decomposition of the matrix Σ_{SS} . When an incremental algorithm is applied, the matrix L_Σ should already be calculated during the previous iteration step. The row vector $L_{k\Sigma}$ and the scalar L_k should be solved in order to incrementally calculate the new Cholesky decomposition $L_\Phi \cdot L_\Phi^T$ for the matrix Φ_k in (20).

Combining (18) and (22) yields the following linear equations:

$$L_\Sigma L_{k\Sigma}^T = \Sigma_{Sk} \quad (23)$$

$$L_{k\Sigma} L_{k\Sigma}^T + L_k^2 = \Sigma_{kk}. \quad (24)$$

In (23), the matrix L_Σ is a lower triangular matrix, since $L_\Sigma \cdot L_\Sigma^T$ is the Cholesky decomposition of Σ_{SS} . Therefore, the vector $L_{k\Sigma}$ can be easily solved from (23) by forward substitutions. Once $L_{k\Sigma}$ is found, the scalar L_k can be directly calculated from (24). After L_Σ , $L_{k\Sigma}$ and L_k are known, the lower triangular matrix L_Φ in (21) is determined.

Next, given the Cholesky decomposition $L_\Phi \cdot L_\Phi^T$ in (20), the matrix inverse Φ_k^{-1} is calculated by solving the following linear equations:

$$L_\Phi \cdot P = I \quad (25)$$

$$L_\Phi^T \cdot \Phi_k^{-1} = P \quad (26)$$

where I denotes an identity matrix. Since L_Φ and L_Φ^T are lower and upper triangular matrices respectively, the linear equations in (25)-(26) can be easily solved by forward and backward substitutions, resulting in the inverse matrix Φ_k^{-1} that we need. As Algorithm 1 proceeds to the next iteration step, the aforementioned procedure will be repeatedly applied and the matrices Φ_k , L_Φ and Φ_k^{-1} will be further updated incrementally.

3.3 Summary

Algorithm 2: Incremental Sensor Selection

1. Start from two given vectors X_S (containing the random variables that have already been selected) and $X_{\bar{S}}$ (containing the random variables that have not been selected), the covariance matrix Σ_{SS} for all random variables belonging to the vector X_S , the Cholesky decomposition $\Sigma_{SS} = L_\Sigma \cdot L_\Sigma^T$, and the inverse matrix Σ_{SS}^{-1} .
2. For each random variable x_k in the vector $X_{\bar{S}}$, calculate the determinant $\det(\Sigma_{kk} - \Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk})$ where Σ_{kS} , Σ_{Sk} and Σ_{kk} are defined in (18).
3. Select the optimal random variable x_k with the maximum determinant value.
4. Given the optimal random variable x_k , solve the row vector $L_{k\Sigma}$ and the scalar L_k from the equations in (23)-(24).
5. Form the matrix L_Φ in (21) and the Cholesky decomposition $L_\Phi \cdot L_\Phi^T$ for the matrix Φ_k in (20).
6. Calculate the matrix inverse Φ_k^{-1} by solving the linear equations in (25)-(26).
7. Remove x_k from the vector $X_{\bar{S}}$ and add it to the vector X_S . Let $\Sigma_{SS} = \Phi_k$, $L_\Sigma = L_\Phi$ and $\Sigma_{SS}^{-1} = \Phi_k^{-1}$.

Algorithm 2 summarizes the numerical method that is developed for incremental sensor selection. It should be used to select the optimal random variable x_k from the vector $X_{\bar{S}}$ at Step 3 and Step 4 of Algorithm 1. As shown in Algorithm 2, the proposed numerical computation starts from two given vectors X_S and $X_{\bar{S}}$. For each random variable x_k in the vector $X_{\bar{S}}$, it calculates the determinant $\det(\Sigma_{kk} - \Sigma_{kS} \Sigma_{SS}^{-1} \Sigma_{Sk})$. The optimal random variable x_k with the maximum determinant is selected. Next, the Cholesky decomposition $\Phi_k = L_\Phi \cdot L_\Phi^T$ and the matrix inverse Φ_k^{-1}

are calculated. Finally, the optimal random variable x_k is moved from the vector X_S to X_S and the matrix Σ_{SS} is replaced by Φ_k , before the next iteration step of Algorithm 1 starts. Relying on these incremental matrix updates, the numerical operations required by Algorithm 2 can be performed with low computational cost. As will be demonstrated by our experimental examples in Section 4, the proposed fast algorithm (i.e., Algorithm 2) is able to achieve 69× runtime speedup over a simple implementation without using incremental updates.

4. EXPERIMENTAL RESULTS

In this section, we demonstrate the efficiency of the proposed sensor allocation and thermal monitoring method by using several experimental examples with both simulation and measurement data. All numerical experiments are run on a 2.53 GHz Linux server with 16 GB memory.

4.1 Simulation Experiment

We first set up a simulation study by using an 8-core microprocessor designed in a 45 nm process. The die area of this microprocessor is 2×2 cm². Its workload is simulated by using 11 SPEC2000 benchmarks (7 integer benchmarks and 4 floating point benchmarks). Wattch [19] is used to estimate the power consumption of each microarchitecture unit (e.g., branch predictor, integer queue, etc.). Next, given the microprocessor floorplan, HotSpot [4] is used to run thermal simulation to generate a set of full-chip thermal maps with the resolution of 128×128 . For testing and comparison purpose, these thermal maps are partitioned into two non-overlapped groups: the training set and the testing set. The thermal maps in the training set are used to calculate the mean value μ_X and the covariance matrix Σ_X in (2), while those in the testing set are used to calculate the error for the proposed thermal map estimation.

We implement four different methods to determine the sensor allocation: (a) the clustering method [10], (b) the partition method [11], (c) the Bayesian method [12], and (d) the proposed entropy method. Once the sensor locations are determined, the maximum-likelihood estimation method in Section 2.2 is used to estimate the full-chip thermal map for any given temperature data collected at the sensor locations. By studying the accuracy of the estimated thermal maps, the efficacy of different sensor allocation algorithms is compared. In addition, to consider the measurement error of each temperature sensor, a small random noise is added to each sensor reading. In this example, the sensor error is modeled as a Gaussian distribution where the mean value is 0 °C and the standard deviation is 1 °C.

Figure 1 shows the temperature estimation error for four different sensor allocation algorithms. Here, the estimation error is calculated by using 5000 thermal maps in the testing set:

$$Error = \frac{1}{\sqrt{5000}} \cdot \sqrt{\sum_{k=1}^{5000} \max_i (\tilde{x}_{i,k} - x_{i,k})^2} \quad (27)$$

where $\tilde{x}_{i,k}$ and $x_{i,k}$ represent the exact and the estimated temperature values at the i th spatial location of the k th thermal map, respectively. Studying Figure 1, we would have two important observations. First, the proposed entropy method outperforms other traditional approaches. Our entropy method achieves 1.4× error reduction, when 9 sensors are selected. Second, the estimation error of the entropy method does not quickly decrease, if we use more than 9 sensors. In other words, deploying 9 sensors for thermal monitoring is a good choice to

explore the trade-offs between estimation accuracy and design complexity in this example.

Figure 2 further plots the spatial locations of 9 temperature sensors selected by different algorithms. In this example, the proposed entropy method arranges all sensors in a regular pattern. Namely, 8 sensors are optimally located at the hotspots of every core, and the other sensor is located at the center of the die to measure the environmental temperature variation. The other three traditional methods, however, cannot find the optimal sensor locations, as shown in Figure 2 (a)-(c). This is the reason why the traditional methods result in large estimation error in Figure 1. In this example, while the optimal sensor allocation is trivial, it provides an excellent test case for us to extensively compare different sensor allocation algorithms.

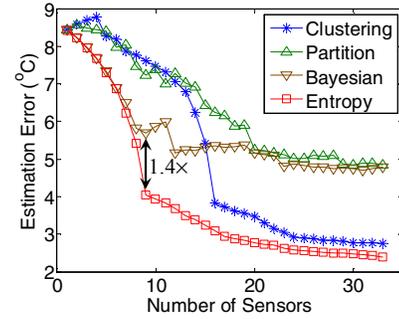


Figure 1. Full-chip thermal estimation error is compared for four different sensor allocation algorithms.

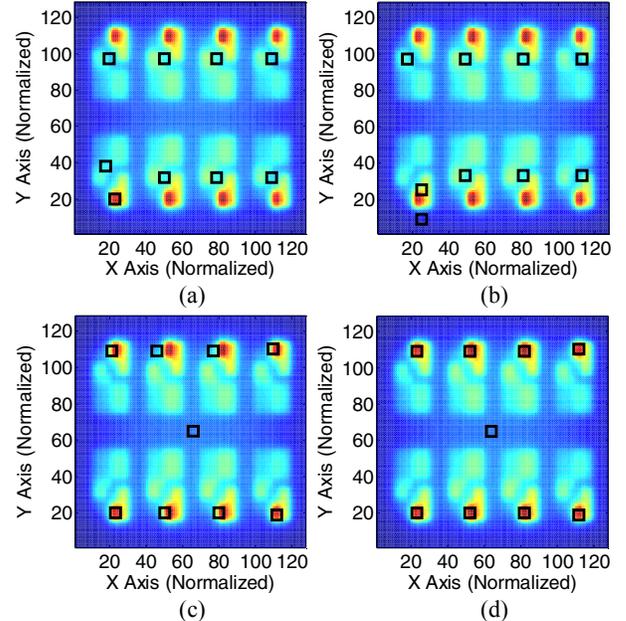


Figure 2. Temperature sensor locations are selected by different allocation algorithms: (a) the clustering method, (b) the partition method, (c) the Bayesian method, and (d) the entropy method.

Clustering [10]	Partition [11]	Bayesian [12]	Entropy (Simple)	Entropy (Fast)
18.78	0.74	7.42	160.24	2.31

Finally, Table 1 shows the runtime for different methods to determine the sensor locations for our experiment in Figure 1.

Note that our proposed fast algorithms achieve $69\times$ runtime speedup over the simple implementation without using these fast algorithms. While the problem size in this example is small (i.e., only 33 sensors are selected in Figure 1), the runtime speedup of the proposed fast algorithms would be more pronounced when applied to larger-size problems.

4.2 Measurement Experiment

In this sub-section, we further demonstrate the efficacy of the proposed entropy method by collecting the temperature sensor data from an industrial dual-core microprocessor that contains 24 temperature sensors distributed in both cores and caches. In this experiment, all 24 temperature sensors are first calibrated using infrared imaging. Next, temperature readings are recorded from these 24 sensors when running a subset of SPEC2006 benchmarks. Similar to the previous example, the measurement data are partitioned into two non-overlapped groups: the training set and the testing set. Our objective in this example is to identify the most important temperature sensors out of these 24 candidates. Once the important sensors are found, the temperature values of other sensors can be estimated and, hence, these unimportant sensors may be eliminated in future design to reduce silicon area and/or power consumption.

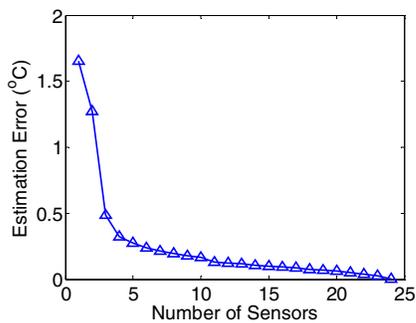


Figure 3. Thermal estimation error decreases, as the number of selected sensors increases.

Figure 3 shows the thermal estimation error as a function of the number of selected temperature sensors. Here, the estimation error is calculated by (27) using the measurement data in the testing set. Note that the estimation error is about $0.3\text{ }^{\circ}\text{C}$, when 5 sensors are selected. Selecting more than 5 sensors has diminishing returns in information collected. It, in turn, demonstrates that the optimal choice is to deploy 5 sensors for thermal monitoring in this example. The aforementioned analysis has potential benefits in both system and design areas. First, it reduces the number of sensors and information processing for thermal-aware OS (i.e., operating system) task scheduling [5]. Second, it demonstrates that our proposed sensor allocation technique is effective in real world and can be used for future microprocessor design.

5. CONCLUSIONS

In this paper, we propose a new information-theoretic framework to efficiently find the optimal spatial locations of on-chip temperature sensors for full-chip thermal monitoring. The key idea is to quantitatively model the uncertainty of on-chip temperature variation by differential entropy. Our experimental results demonstrate that the proposed entropy-based method achieves superior accuracy (around $1.4\times$ error reduction) for full-

chip thermal monitoring over other traditional methods. The techniques developed in this paper can be further incorporated into various thermal management schemes for hotspot-limited microprocessors.

6. ACKNOWLEDGEMENTS

The authors acknowledge the support of the C2S2 Focus Center, one of six research centers funded under the Focus Center Research Program (FCRP), a Semiconductor Research Corporation entity. This work is also supported in part by the National Science Foundation under contract CCF-0915912.

7. REFERENCES

- [1] H. Hamann, A. Weger, J. Lacey, Z. Hu and P. Bose, "Hotspot-limited microprocessors: direct temperature and power distribution measurements," *IEEE JSSC*, vol. 42, no. 1, pp. 56-65, 2007.
- [2] M. Pedram and S. Nazarin, "Thermal modeling, analysis, and management in VLSI circuits: principles and methods," *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1487-1501, 2006.
- [3] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," *IEEE HPCA*, pp. 171-182, 2001.
- [4] K. Skadron, M. Stan, H. Wei, S. Velusamy, K. Sankaranarayanan and D. Tarjan, "Temperature-aware microarchitecture," *IEEE ISCA*, pp. 2-13, 2003.
- [5] J. Choi, C. Cher, H. Franke, A. Weger, and P. Bose, "Thermal-aware task scheduling at the system software level" *IEEE ISLPED*, pp. 213-218, 2007.
- [6] Y. Wang, K. Ma and X. Wang, "Temperature-constrained power control for chip multiprocessors with online model estimation," *IEEE ISCA*, pp. 314-324, 2009.
- [7] T. Ebi, M. Faruque and J. Henkel, "TAPE: Thermal-aware agent-based power economy multi/many-core architectures," *IEEE ICCAD*, pp. 302-309, 2009.
- [8] C. Cher and E. Kursun, "Exploring the effects of on-chip thermal variation on high-performance multicore architectures," *ACM TACO*, vol. 8, no. 1, pp. 2:1-2:22, 2011.
- [9] S. Bokar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi and V. De, "Parameter variations and impact on circuits and microarchitecture," *IEEE DAC*, pp. 338-342, 2003.
- [10] S. Memik, R. Mukherjee, M. Ni and J. Long, "Optimizing thermal sensor allocation for microprocessors," *IEEE TCAD*, vol. 27, no. 3, pp. 516-527, 2008.
- [11] A. Nowroz, R. Cochran and S. Reda, "Thermal monitoring of real processors: techniques for sensor allocation and full characterization," *IEEE DAC*, pp. 56-61, 2010.
- [12] Y. Zhang, B. Shi and A. Srivastava, "A statistical framework for designing on-chip thermal sensing infrastructure in nano-scale systems," *IEEE ISPD*, pp. 169-176, 2010.
- [13] F. Zanini, D. Atienza, C. Jones and G. Micheli, "Temperature sensor placement in thermal management systems for MPSoCs," *IEEE ISCAS*, pp. 1065-1068, 2010.
- [14] S. Sharifi and T. Rosing, "Accurate direct and indirect on-chip temperature sensing for efficient dynamic thermal management," *IEEE TCAD*, vol. 29, no. 1, pp. 1586-1599, 2010.
- [15] Y. Zhang, A. Srivastava and M. Zahran, "Chip level thermal profile estimation using on-chip temperature sensors," *IEEE ICCD*, pp. 432-437, 2008.
- [16] R. Cochran and S. Reda, "Spectral techniques for high-resolution thermal characterization with limited sensor data," *IEEE DAC*, pp. 478-483, 2009.
- [17] Y. Zhang and A. Srivastava, "Accurate temperature estimation using noisy thermal sensors," *IEEE DAC*, pp. 472-477, 2009.
- [18] W. Zhang, X. Li and R. Rutenbar, "Bayesian virtual probe: minimizing variation characterization cost for nanoscale IC technologies via Bayesian inference," *IEEE DAC*, pp. 262-267, 2010.
- [19] D. Brooks, V. Tiwari and M. Martonosi, "Watch: a framework for architecture-level power analysis and optimizations," *IEEE ISCA*, pp. 83-94, 2000.
- [20] C. Bishop, *Pattern Recognition and Machine Learning*, Prentice Hall, 2007.
- [21] R. Horn and C. Johnson, *Matrix Analysis*, Cambridge Press, 2007.