A Statistical Framework for Designing On-chip Thermal Sensing Infrastructure in Nano-scale Systems*

Yufu Zhang  
Dept. of ECE  
University of Maryland  
College Park, MD 20743, USA  
yufuzh@umd.edu

Bing Shi  
Dept. of ECE  
University of Maryland  
College Park, MD 20743, USA  
bingshi@umd.edu

Ankur Srivastava  
Dept. of ECE  
University of Maryland  
College Park, MD 20743, USA  
ankurs@umd.edu

ABSTRACT
Thermal/power issues have become increasingly important with more and more transistors being put on a single chip. Many dynamic thermal/power management techniques have been proposed to address such issues but they all heavily depend on accurate knowledge of the chip’s thermal state during runtime. In this paper we describe a unified statistical framework for designing an on-chip thermal sensing infrastructure which can be used to track the chip’s thermal state at runtime. Specifically we address the following problems: (1) sensor placement; (2) sensor data compression; (3) sensor data fusion; (4) overall interplay. Our methods exploit the thermal correlation to generate the overall solution. To ensure both the noiseless and noisy sensor settings, our framework is also capable of choosing the appropriate degree of compression for each sensor while accounting for their local space constraints when doing the sensor deployment. The experimental results showed that our infrastructure can improve the temperature estimation accuracy by 27% (on average) as compared to an equivalent system that uses range-based placement and uniform compression. It took our methods about 6.3 seconds to decide the overall solution for placement, compression and data fusion at design stage. This demonstrates the effectiveness and applicability of our unified statistical design methodology.

Categories and Subject Descriptors
B7.2 [Integrated Circuits]: Design Aids
General Terms
Algorithms, Design, Reliability
Keywords
Temperature, Sensor placement, Statistical, Estimation

1. INTRODUCTION
Due to the effects of technology scaling and the increasing operating frequency of the modern high performance processors, the thermal and power problems are becoming more critical every day. Many dynamic thermal management techniques (DTM) try to address such problems by throttling voltage and/or operating frequency in exchange for less power dissipated. Such DTM schemes rely heavily upon accurate knowledge of the system thermal state in order to come up with the best thermal control decision. To address this problem, several on-chip thermal sensor placement algorithms have been proposed in [1, 2, 3] to systematically deploy sensors across the chip. Although such work is promising, very few researchers have investigated the development of a complete framework that enables accurate and efficient thermal sensing and estimation. In this paper we propose a general statistical framework as well as a unified methodology for designing the on-chip thermal sensing infrastructure.

On-chip thermal sensing infrastructure consists of several important design components which have strong interplays among each other. These components include sensor placement, individual sensor design/compression and data fusion (as shown in figure 1). Let us first take a look at the normal information flow in any sensor network infrastructure. First the sensors will collect information which reflects its local environment (temperature in our case). The data collected by each sensor will undergo preliminary processing/compression (digitalization for example). When we have local resource constraints such as area and transmission overhead, data compression at each sensor is highly desirable. In addition, sensors (especially in on-chip environments) can be particularly susceptible to noise, therefore the sensor data processing/compression must be conducted judiciously to account for such effects. The compressed sensor readings will then be collected at a data fusion center to generate a complete thermal profile of the entire system. Within the fusion center, all sensor readings will be stored in a central register. A high-level coordinator (which could implemented either in hardware or software) will take these sensor readings for information extraction purposes (eg. signal filtering, noise removal and information estimation). The final outcome would be accurate knowledge of the thermal state of the entire chip.

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.

*This work is partly supported by NSF under grants CCF-0728969 and CCF-0917057.

ISPD '10, March 14–17, 2010, San Francisco, California, USA.
Copyright 2010 ACM 978-1-60558-920-6/10/03 ...$10.00.
Partly due to the nature of the thermal sensing problem and partly due to global area/power constraints, each of these design components are strongly interdependent. For example, the size of the central register, which depends upon the amount of area available, will impact the number of sensors we can place and the degree of compression necessary at each sensor. The local area slack to fit individual sensors is another constraint. It may be easier to fit several sensors with high compression ratios than a few sensors with no compression. The overall complexity of this sensor infrastructure can be determined as a tradeoff between the degree of accuracy desired and the implementation overhead.

In this paper, we develop a unified statistical methodology for instantiating such a thermal sensing infrastructure: we decide sensor locations, degree of sensor compression and the design of fusion center in a unified way. Our methodology is capable of enforcing constraints such as the area available for each design component and also accounts for the intricate interplay between them. The specific contributions of this paper are as follows:

1. **Sensor placement**: We develop a statistical methodology for deciding the sensor locations. Most existing works on this topic assign a range for each sensor [2, 1, 3] and allocate sensors so as to cover all the potential hotspots. The metrics of range and cover are inaccurate for thermal sensors since sensors measure only the temperature of the location where they are placed (unlike cameras which have a field of view). Instead, our method, uses the statistical correlations between sensor and hotspot temperature to predict the probability of capturing all the hotspots. This probability, which is a more sound metric than range, is used to drive sensor placement.

2. **Sensor design and compression**: Due to the potential lack of space to fit thermal sensors and the associated wiring, local compression of data is necessary. Compression of sensor data is also needed due to limited space for storing the sensor readings in the central register at the fusion center. Sensors are also prone to noise caused by fab-variability, supply voltage fluctuation etc. Our sensor design and compression methodology accounts for all the above considerations. To achieve our goals we use concepts from compression, signal estimation theory, optimization and VLSI design.

3. **Fusion center design**: Given compressed and possibly noise-corrupted sensor observations, we develop statistical techniques to estimate the chip thermal profile by exploiting the thermal correlations among different chip modules.

4. **Exploiting the interdependency**: In order to obtain the best thermal sensing infrastructure solution which also has the minimal overhead (area, power etc), we will exploit the interplay between the above-mentioned aspects. For example, while deciding the sensor locations, our method will avoid areas where fitting sensors is difficult. Also, the finite number of bits available at the central register to store all sensor readings should be a limiting factor when deciding the total number of sensors allocated on chip and their compression rates. When designing the sensor placement schemes we introduce a feedback mechanism for incorporating the effects of compression.

To the best of the our knowledge, such a complete and unified methodology for designing the on-chip thermal sensing infrastructure has not been investigated in the past. To demonstrate the effectiveness of our methods, we did experiments assuming the sensors are either noiseless (in the ideal scenarios) or noisy (in the realistic situations). Our results showed that having several sensors with compressed observations outperformed having a few sensor with no compression when given the same space constraint at the global fusion center. On average our sensor placement and compression schemes can achieve about 27% reduction in the overall RMS error as compared to the range-based placement scheme and uniform compression (with about equivalent overhead). Our algorithms only took around 6 seconds in the worst case to develop the overall solution for placement, compression and data fusion. It is also noteworthy that our framework is general enough to incorporate different statistical models.

The rest of the paper is organized as follows: in section 2 we introduce our statistical methodology for co-designing the sensor and fusion center while accounting for their interplay. We then present our sensor placement algorithm and the complete design flow in section 3 and 4 respectively. The experimental results are reported in section 5 and the paper is concluded in section 6.

## 2. SENSOR & FUSION CENTER CO-DESIGN

### 2.1 Fusion Center

A global fusion center collects and combines the sensor readings in order to estimate the chip's thermal state at any given time. It has two distinct components: central register and fusion algorithm. The central register is basically a register that holds all the thermal sensor readings (it could be a single or a combination of several actual registers). The fusion algorithm utilizes the combined sensor observations to estimate the complete thermal profile of the chip [4]. The design of the entire thermal sensing infrastructure depends critically on how a few thermal sensor readings are used to predict the thermal profile. In this paper we use a variant of the statistical approach presented in [4]. This approach combines the information provided by a few on-chip thermal sensors with the thermal statistical information (such as the thermal correlations among different chip locations) to generate accurate temperature estimates at all chip locations. The idea is that by exploiting the thermal correlations that existed among different chip modules, the temperature sampled at the sensor locations can be used to predict the thermal state at other chip locations as well. We will describe this fusion algorithm in more detail below.

![Figure 2: A simplified chip from the thermal perspective](image)
the chip temperatures [4, 7, 8]. Also extensive experiments and/or simulation can be performed to obtain such required statistical information. When the randomness in the temperature vector $\vec{T}$ exhibits Gaussian nature the conditional expectation is the optimal estimator for $\vec{T}$ (the complete thermal profile) when given sensor observations $T_s$ (see [9] for detailed proof). The analytical form of this estimator is presented in equation 1. Note that since there are many independent sources of randomness, their collective effects are reasonably close to a Gaussian distribution (based on the central limit theory). Hence in practice equation 1 can be expected to generate accurate thermal estimates. Also note that the thermal correlations exist due to physical proximity and also similar power behaviors of different functional modules (for details please see [4]). In the following formulation such correlations are captured in the covariance matrix $\Sigma_{TT}$.

$$E(\vec{T}|\vec{T}_s) = \mu_s + \Sigma_{TS}\Sigma_{SS}^{-1}(\vec{T}_s - \mu_s)$$

$$\Sigma = \Sigma_{TT} - \Sigma_{TS}\Sigma_{SS}^{-1}\Sigma_{ST}$$

(1)

(2)

Here the thermal sensors are assumed to be placed in some of the grid cells (say, subset S). Therefore, at runtime, we can observe the temperatures at these grids ($\vec{T}_s$) by simply sampling the sensors. $\Sigma_{SS}$ represents the covariance matrix for the sensor grids and it is a submatrix of $\Sigma_{TT}$ where each row/column in $\Sigma_{SS}$ corresponds to a grid where a sensor has been placed. By the same logic, $\Sigma_{TT}$ is the covariance between all grids (represented by set T) and sensor grids (set S). Also, $\mu_s$ is the mean temperatures of sensor grids S. It is noteworthy that equation 1 exploits the covariance (and hence correlation) between sensor grids and other chip grid to perform the estimation. Equation 2 gives the new covariance matrix (or conditional covariance) of $\vec{T}$, now that some temperatures at sensor locations are known. Note each diagonal element of this new covariance matrix $\Sigma$ indicates the new reduced variance of the corresponding grid temperature once given the sensor information. This variance measures the fundamental uncertainty in the thermal estimates generated by equation 1 and depends on the sensor locations, i.e. the selection of set S and hence $\Sigma_{TS}/\Sigma_{SS}$ (we discuss more on sensor placement in subsequent sections). It is noteworthy that this statistical approach is more sound than the existing range-based approaches [1, 2, 3]; instead of assuming a range for each sensor and ignoring the thermal gradient within such range (and also discarding the information outside the range), it calculates the conditional expectation of the temperatures at all grid locations. It also gives the variance associated with such estimates which reflects our confidence in our estimated temperatures. From the expectation and variance we can easily deduce the possibility of capturing the potential hotspots. The above presented fusion algorithm works by first combining all sensor readings into a central register information. Then the fusion center exploits the covariance matrix $\Sigma$ and then combines the multiple thermal estimates based on the value in this register. Equation 1, which estimates the thermal profile after obtaining sensor readings $T_s$, could be implemented in the operating system.

2.2 Noisy Thermal Sensor

In this section, we focus on thermal sensor design and compression. To make things more concrete, we use ring oscillator based thermal sensor as the example to describe our method. Note that our methodology is general and can be applied to any other type of sensors equally well. A ring oscillator (RO) simply consists of an odd number of inverters. The change in temperature will affect the delay of each inverter and hence change the frequency of the RO. We can have a counter at the output of the RO to count the number of state flips within a fixed period of time $t_p$ (see figure 3). The output of this counter at the end of the counting period reflects the frequency of the RO. Due to the fact that the frequency of a RO has a close-to-linear relationship with its local temperature, ROs are often used to implement thermal sensors. The number of bits needed to represent the counter output captures the precision of the sensor.

![Figure 3: Ring oscillator as a thermal sensor](image)

Figure 3: Ring oscillator as a thermal sensor

In the ideal case where a sensor is noiseless it gives the same sensor reading for the same grid temperature irrespective of its location and time of sampling. In such a case the sensor readings present no ambiguity whatsoever and could be relied upon completely. Since the number of bits b of the counter output is fixed, the sensor cannot provide infinite accuracy. If we uniformly divide the temperature range $H_{total}$ into $n = 2^b$ intervals $\{H_1, H_2, \ldots, H_n\}$ where each $H_i$ represents a temperature sub-range, then the sensor can report the specific sub-range $H_i$ that the sensor grid is experiencing. The finite size of the counter imposes a small quantization error.

In reality, thermal sensors are highly susceptible to fabrication variability, supply voltage fluctuations etc [6, 10]. To understand the effect of such sensor noise we used Monte Carlo simulation and obtained the randomness in the RO frequency (figure 4) caused due to fabrication variability etc. We assumed 5% variation in threshold voltage, channel length/width, oxide thickness and supply voltage. As shown, for each underlying actual temperature, the observed sensor frequency can take a range of values.

![Figure 4: Simulated RO frequency distribution (10^7 samples each) under fabrication randomness for different underlying temperatures ranging from 20°C to 100°C with 20°C increment](image)

Figure 4: Simulated RO frequency distribution (10^7 samples each) under fabrication randomness for different underlying temperatures ranging from 20°C to 100°C with 20°C increment

Hence, given a sensor reading, the actual temperature that caused the reading cannot be decided deterministically. In order to estimate the temperature for a given noisy thermal reading, we formulate the problem in a hypothesis testing framework [9]. Hypothesis testing has the advantage that it can generate accurate results for any type of noise distribution. It is also general enough to be easily extended to handle the sensor compression and the bit allocation problems as well (which we will discuss in later sections).

As mentioned earlier, we can divide the thermal range $H_{total}$ into $n = 2^b$ sub-ranges $\{H_1, H_2, \ldots, H_n\}$. We are only interested in estimating which sub-range the sensor temperature falls within. We can assume each $H_i$ is a hypothesis for the underlying temperature of this sensor with an associated prior probability ($P_i$ for hypothesis $H_i$). Prior probability captures the probability of having sub-range $H_i$ as the temperature at the sensor grid and can be obtained
by simulating benchmarks or typical chip workloads. For a
given thermal hypothesis $H_i$ at the sensor grid, the reading
from this sensor is actually random due to sensor noise
and fab-variability. The randomness of the noisy sensor ob-
servation for each underlying hypothesis can be modeled as
a probability density function (abbr. PDF) as illustrated
in figure 4. This modeling could be obtained using various
statistical schemes that characterize the behaviors of the
thermal sensor under different thermal conditions. Given a
sensor observation $T_o$, our goal is to choose one of the hy-
potheses as our best prediction for the actual sensor temper-
aturce such that the expected prediction error is minimized.

Formal problem formulation: Let us define $P_i$ as the
prior probability of hypothesis $H_i$. Note that $\sum_{i=1}^{n} P_i = 1$
since one of these $n$ hypothesis must be true. Now, given the
set of all observations $\Omega$ (which for a thermal sensor with $b$
output bits would contain $2^b$ values), we would like to parti-
tion $\Omega$ into $n$ (number of hypotheses) subsets $\{\Gamma_1, \Gamma_2, \ldots, \Gamma_n\}$
such that each subset corresponds to a specific hypothesis
prediction (if the observation falls within a certain subset
it means the corresponding hypothesis must be true). The
challenge is to come up with this partition/decision rule $\delta$
such that the expected prediction error is minimized.

The prediction error is simply the absolute differ-
ence between the real and the predicted hypothesis
prediction (if the observation falls within a certain subset
it means the corresponding hypothesis must be true). The
expected penalty as in

\[ E(\|H_p - H_{real}\| | T_o) = \sum_{i=1}^{n} \|H_o - H_i\| \times \text{prob}(H_{real} = H_i | T_o) \]

where $H_{real}$, $T_o$ and $H_p$ represent the real hypoth-
thesis (basically real temperature), the noisy sensor observation
and our prediction respectively. The equation itself is quite
self-explanatory. Since the sensor is noisy, for a given ob-
servation $T_o$, we can have different hypotheses that could
have caused it. This fact is captured by the probability
\[ \text{prob}(H_{real} = H_i | T_o) \] which could be computed as follows:

\[ \text{prob}(H_{real} = H_i | T_o) = \text{prob}(T_o | H_{real} = H_i) \times P_i \]

Bayes-Rule

Here $\text{prob}(T_o | H_{real} = H_i)$ is simply the probability of ob-
taining a specific thermal reading for a given real sensor
temperature (hypothesis) and could be computed a-priori
using statistical schemes (for example as we illustrated
in figure 4). Based on equation 6 we can easily compute equa-
tion 7. Now the overall expected cost associated with the
decision rule can be defined as follows:

\[ E(\|H_p - H_{real}\| | T_o) = \sum_{X \in \Omega} E(\|H_p - H_{real}\| | T_o) \cdot \text{prob}(T_o) \]

Then the optimal prediction for each observation ($T_o \rightarrow H_p$)
could be stored in a look up table.

\[ \delta(T_o) = \arg\min_{H_p \in \Omega} E(\|H_p - H_{real}\| | T_o) \]

The above decision rule can be pre-determined quite easily
for all possible sensor observations given the required sta-
tistical information. Thus our decision rule can be easily
implemented in hardware as an encoder stage of the sensor
as shown in figure 3. Essentially this encoder stage is re-
ponsible for translating the noisy sensor readings into our
predictions.

2.3 Fusion Center & Sensor Co-Design

Now, we extend our hypothesis testing based approach for
coop-design of the fusion center and the sensors.

2.3.1 Problem Formulation

For clarity of exposition let us first suppose the thermal
sensors are noiseless (the noisy sensor case is similar and
will be discuss later). The fusion center design problem es-
tentially boils down to how to allocate the total number of
bits in the central register to all sensors. A larger central
register implies more bits in the thermal sensors which in
turn implies more precise sensor readings and more accu-
rated overall estimates (assuming a fixed sensor placement).
However more bits in the central register would also imply
more area overhead for storing these bits and also commu-
nicating more data from sensors thereby complicating the
wiring. Note that the exact implementation of the central
register (one big or many small actual registers) is not of
concern here. We are just trying to address the finite total
space of storing sensor data. In this section, we assume that
the number of sensors and their placement is fixed so we
focus primarily on the central register size.

For a given size of $M$ bits, we can have several policies
for distributing them among the sensors. A uniform policy
would imply the same level of accuracy for each sensor. In-
terestingly, some sensors are more informative than others
(ones that have a higher correlation with hotspot locations)
and therefore should be given more precision (number of
bits). Also, some sensors, if given more bits, may not have
the space available for routing the extra wires to the cen-
tral register. These considerations need to be accounted for
while distributing the total $M$ bits. The choice of $M$ it-
self is more complex since it depends critically on how much
area/power penalty we can tolerate, how much accuracy is
desirable and sensor locations as well. Hence $M$ can be spec-
died by designers based on various design goals.

Let us suppose each sensor $i$ can have at most $b_i$ bits.
Here $b_i$ depends on the space available for the sensor and
its wiring. Based on the informativeness of the sensors and
their space constraints (parameter $b_i$), we would like to dis-
tribute the total $M$ bits to sensors such that the overall
thermal estimates (given by equation 1) are as accurate as
possible. Let $\tilde{T}_i^a$ and $\tilde{T}_i^b$ represent the accurate (without $M$
constraint) and the compressed (due to a finite $M$) sensor
observations respectively, the thermal estimates under these
two situations can be obtained using equation 1 as $E(\tilde{T}_i^a)$
and $E(\tilde{T}_i^b)$ respectively. Note that they represent a vector of
thermal estimates where the $i$-th element is the estimate for
the $i$-th grid. Thus the error in the $i$-th grid can be de-
ned as $E(\|\tilde{T}_i^a - \tilde{T}_i^b\|)$. Now we would like to allocate the total $M$ bits to sensors such that the overall expected
error is minimized. Our problem can be formulated as the
following optimization problem:

\[ \text{minimize } E(\text{error}(s_1, s_2, \ldots, s_n)) \]

s.t. \[ 0 \leq s_i \leq b_i \]
\[ \sum_{i=1}^{n} s_i \leq M \]
Here \( s_i \) is the number of bits assigned to sensor \( i \) and \( b_i \) is the maximum number of bits for this sensor as described earlier. Note that \( b_i \) itself is a random variable since the underlying observations are random. Hence the expectation of the error is used as the cost function for our optimization problem. If \( M \) was not a limiting constraint and all sensors worked at their perfect precision (i.e. \( s_i = b_i \) for \( \forall i \)), then the expected error should be zero. Else the error creeps in due to compression of the sensor observations. For a fixed sensor placement, this error depends explicitly on how the total \( M \) bits is allocated among sensors and implicitly on how each sensor is compressed locally. Hence allocation of bits to sensors and designing sensors (so that the allocated bits are used most effectively) must be done hand in hand. Now let us consider the error term below:

\[
\text{error}(s_1, s_2, \ldots, s_n) = \sum_{\forall \ell \neq c \in s} (|E(T_{\ell c} - T_{\ell c}^k)| - E(T_{\ell c}^k))
\]

(11)

\[
= \sum_{\forall \ell \neq c \in s} (\Sigma_{x \in S} \Sigma_{y \in S} (T_{\ell c}^k - T_{\ell c}^n))
\]

(12)

\[
= \sum_{\forall \ell \neq c \in s} (\Sigma_{x \in S} \Sigma_{y \in S} (|\Delta T_{\ell c}|))
\]

(13)

Here 12 is obtained by substituting equation 11 into 11.

Now let us focus on the term \( E(|T_{\ell c}^k - T_{\ell c}^n|) \). This is the local expected error at the sensor \( k \) caused due to compression. Note that this error depends explicitly on how the total \( M \) bits are allocated among sensors and implicitly on how each sensor is actually compressed. Let us address the second problem first since the first one depends on the second one.

**Sensor Compression:** In principle and form, the expected error \( E(|T_{\ell c}^k - T_{\ell c}^n|) \) can be defined in a hypothesis testing framework similar to the one we described in subsection 2.2 (see eq. 7).

\[
E(|T_{\ell c}^k - T_{\ell c}^n|) = \sum_{\forall \ell \neq c \in s} E(|T_{\ell c}^k - T_{\ell c}^n|) \times \text{prob}(T_c)
\]

(22)

Here subscript \( k \) represents the \( k \)-th sensor. Initially this sensor has \( b_k \) bits (its local maximum). Let us define a \((2^{b_k} \times n)\) matrix \( Q \) where \( n \) is the number of hypotheses (grid temperatures) that the sensor can experience. Each row of this matrix corresponds to a specific observation. The element \( Q[l, j] \) stores the joint probability of getting observation \( l \) when hypothesis \( j \) is true. When the sensor is noiseless, there can only be exactly one hypothesis with non-zero probability for each observation. All other hypotheses have zero probability (note that this can only happen when \( b_k = n \)). Now we compress this sensor by, say, 1 bit. This means that the sensor will lose some fidelity and will have to be compressed from a larger set of fine-grained observations to a smaller set of coarse-grained observations. This implies some rows (observations) in \( Q \) should be combined. Let us suppose that we combine row \( i \) and row \( l \) into one row. Now the probability of getting this new combined observation under hypothesis \( j \) is simply \( Q[i, j] + Q[l, j] \). Thus the compression of sensor observations amounts to combining and adding the corresponding rows. The objective of the compression scheme is to decide which rows to combine and which hypotheses to predict for the new set of compressed observations such that the cost function (22) is minimized. Now, let \( Q' \) be the new matrix after compression which has \( 2^{b_k} \) rows. Note that in our case the cost of predicting hypothesis \( i \) while hypothesis \( j \) is true is simply \(|H_i - H_j|\). Given the compressed matrix \( Q' \), we would like to generate a decision rule that translates each observation into a predicted hypothesis such that the expected cost is minimized. Just like in the noisy sensor case (subsection 2.2), the same sensor reading can be observed under multiple hypotheses. The best decision rule that achieves the minimum cost can be generated using the techniques discussed earlier (basically equation 8). The cost of this optimal decision rule is given by equation 22 and depends on how we compress the sensor observations. This cost can be calculated in the same way as we calculate equation 7 based on 4 – 6. Note that the underlying probabilities needed by these equations can be obtained by calculating the matrix \( Q' \). The compression at each sensor will generate a \( Q' \) from \( Q \) by combining \( 2^{b_k} \) rows into \( 2^{b_k} \) rows such that 22 is minimized. The compression policies generated could be implemented in the encoder stage of the sensor (see figure 3). Given an observation that has \( b_k \) bits, it outputs a new code of length \( s_k \) bits where \( s_k \) is the number of bits allocated to the sensor. Also, several observations that have been compressed into one are given the same output code of length \( s_k \) bits. The details of the algorithm for generating this \( Q' \) are described in Algorithm 1.

**Bit Allocation:** As explained above, when given a fixed number of bits for a certain sensor \( k \), we can come up with a compression scheme as well as a prediction policy for any observation so that the expected local misprediction error at this sensor \( E(|T_{\ell c}^k - T_{\ell c}^n|) \) is minimized. Now we present our scheme for allocating the total \( M \) bits among sensors such that the overall expected error for the entire chip (equation 16) is minimized. This part is relatively easy since in our
problem formulation we have already established the relationship between a sensor's local compression error and the overall error. Therefore, by minimizing the error as highlighted in equation 21. We can start by assuming the maximum bits $s_i = b_i$ for each sensor $i$ and gradually reduce them until the constraint in 10 is met. This can be done in a greedy fashion by reducing one bit at a time at the sensor which results in the least cost increase in each iteration (compression causes increase in cost). Based on the way in which we construct the overall cost function 21, our greedy bit allocation scheme will allocate fewer bits to the less informative sensors while giving more precision to the more important ones.

### 2.3.2 Co-Design Algorithm

Now we present an algorithm for simultaneous fusion center and sensor co-design. Our technique estimates the overall error for any bit allocation scheme by generating the best compression scheme at each sensor and then evaluating equation 21. The results are then used to drive the bit allocation process. The final outcome includes both a bit allocation scheme and a compression policy at each sensor. The algorithm is shown below:

**Algorithm 1:** Sensor & Fusion Center Co-Design:

**Input:** Initially $s_i \leftarrow b_i$ for each sensor $i$

**Output:** Bit allocation and sensor compression schemes

1. **while** $\sum_{i=1}^{n} s_i \geq M$ **do**
   2. Choose sensor $k$ with the least sensitivity $g_k$ to the overall cost (see equation 21)
   3. If $s_k = 0$ then choose the next sensor
   4. Let $Q$ be the matrix for sensor $k$ such that $Q[i,j]$ is the probability of getting observation $i$ with hypothesis $j$
   5. Generate all possible $Q^\text{new}$ (by combining the rows of $Q$) such that the number of rows in $Q^\text{new}$ is 1/2 of $Q$ (one bit reduction)
   6. For each $Q^\text{new}$ select the optimal decision rule using equation 8 and use the associated cost to determine the best $Q^\text{opt}$
   7. $Q \leftarrow Q^\text{opt}; s_k \leftarrow s_k - 1$; update the overall error (equation 21)

2. **end while**

We start with all sensors allocated the maximum bits. Then in each iteration, we select a sensor which has the least sensitivity to the overall expected error and compress it by 1 bit. We repeat this process until we reach a feasible solution that satisfies the global $\text{M}^\text{p}$ constraint.

#### 2.3.3 Noisy Case

When the sensors are noisy, for a given observation there could be several hypothesis possible. Hence, in the uncompressed case, the $Q$ matrix is such that for each observation $i$, the row $Q[i]$ can have several hypothesis with non-zero probabilities. Fundamentally, algorithm 1 can simply take this new $Q$ matrix and perform the compression in a similar fashion as the noiseless case. Hence the same algorithm can be applied to the noisy case as long as the $Q$ matrix is updated appropriately.

### 3. SENSOR PLACEMENT

In this section we describe a thermal sensor placement approach that is very different from most existing placement algorithms [1, 3]. As explained earlier, most existing placement algorithms assume that each sensor has a "coverage region" around it and the temperature within this region can be accurately monitored. Yet the sensor has no or very little knowledge of the temperature outside of this region. Thus the goal is to place as few on-chip sensors as possible so that all pre-identified hotspots fall within the coverage region of a certain sensor. In reality thermal sensors only measure the temperatures of the grids that they are located in. Defining a range on the sensor's temperature, becomes more difficult. On the one hand the thermal gradient within the sensor range is ignored. On the other hand the information about locations outside the sensor coverage region is discarded. Thus the accuracy of the range-based methods highly depends on how large or small this sensor range is. Granted, if the range is chosen to be small it can achieve reasonable accuracy. However, even such accuracy comes with a cost: more sensors need to be placed on chip due to the smaller range each sensor can cover. Our method does not have such drawbacks, it exploits the thermal correlation to place sensors so that the sensors not only provide thermal information for its local area but also provides information for remote locations as long as there exists certain amount of correlation. Thus our methods can provide better thermal sensing accuracy with even fewer sensors. In this paper we use statistical techniques (equation 1) to estimate the entire thermal profile from limited sensor observations. The fundamental error associated with a placement scheme is given by the variance shown in equation 2. The overall error also depends on the degree of compression imposed on the sensors. As discussed earlier, the more spatially constrained a sensor is, the more we will have to compress its readings since we do not have any space available for routing data wires. A sensor placement that has many sensors with high compression factors could be undesirable since either the sensors are less informative (leading to compression in favor of others) or do not have sufficient space. Hence sensor placement is a complex design problem which needs to consider not only how much information a potential sensor location can provide but also the available space slack at that location. It also has to account for the finite size $M$ of the central register.

#### 3.1 Problem Formulation

The purpose of the sensor placement algorithm is to choose optimal locations for a limited number of sensors such that the entire thermal profile (or certain critical regions of interest) can be estimated as accurate as possible.

Let $S$ be a subset of the grids of size $n$ representing the sensor locations. As mentioned earlier, there are two kinds of errors associated with a sensor placement: the fundamental error given by equation 2 and the compression error. A solution with low fundamental error might have high compression error, for example, the sensor locations might not have sufficient space to fit the routing wires. Given a sensor count, $m$, the sensor placement problem is to find the optimal locations for these sensors such that the overall error is minimized. Note that since the compression error accounts for the area constraint at sensor locations and finiteness of the central register, accounting for compression error implies accounting for space limitations.

For simplicity in exposition, let us ignore the compression error for the moment and focus primarily on the fundamental error. Finding a sensor placement that minimizes this error can be formulated as follows:

$$\text{choose } S \subseteq T \text{ with } |S| = m \quad (23)$$

such that $\text{trace}(\hat{\Sigma})$ is minimized

Here $\hat{\Sigma}$ is calculated as in equation 2. Basically we would like to choose a subset $S$ of size $m$ (from all grid locations $T$) as our sensor locations. Based on the choice of $S$, $\hat{\Sigma}$ becomes the new covariance matrix of all grid temperatures. The objective function represents the total variance of each grid for a given choice of $S$ which essentially captures the fundamental error. This is calculated by computing the sum of all the diagonal elements of $\Sigma$ (trace of $\Sigma$). Note that the diagonal element of a covariance matrix indicates the variance of the random variable represented by the corresponding row. Optimizing this cost function is a very complex task and the
problem is in general NP-hard. Intuitively, we would like to select those sensor locations that provide the maximum information about places where sensors do not exist. Also, in order to minimize redundancy, the sensors themselves should have little information about each other. The information content is basically predicted by the correlation. If a sensor location is highly correlated with other grids then in general it is a good candidate. If a sensor location has high correlation with other sensors and lower correlation with other non-sensor location, then it could be removed. Since the real cost function above is very complex, we simplify the cost function as follows:

\[
\sum_{\forall y \in y_{grid}} \max \left( 0, 1 - \sum_{\forall sensor \in y} c_{ij}^2 \right)
\]

(25)

Here \(c_{ij}\) is the correlation between \(i\) and \(j\). For each grid location \(i\), the expression \(\max(0, 1 - \sum_{\forall sensor \in i} c_{ij}^2)\) captures the amount of information provided by sensors about \(i\). If a lot of sensors have high correlation with \(i\), then this expression would evaluate to almost 0. Note that if \(i\) is a sensor location itself, then this expression would always evaluate to 0 since \(c_{ij} = 1\). Choosing a set of sensors that maximizes the correlation between sensors and non-sensors will be encouraged by this cost function. On the other hand, due to the inherent max function, it will avoid picking too many sensors that have high correlation with the same grid (this could indicate redundancy). This cost function will also discourage choice of sensors which have high correlation among each other (which is wasteful). This is because the cost of a sensor grid \(i\) is 0 regardless of the presence of other sensors. Hence a judicious sensor placement algorithm will not put more sensors to improve the prediction accuracy at another sensor location \(i\) since its contribution to the overall cost function is 0 anyway. The experimental results presented subsequently will illustrate the effectiveness of this cost function.

The overall problem formulation (when we ignore the compression error) is basically choosing a set of \(m\) sensor locations such that the objective function in equation 25 is minimized. The technique for incorporating the compression error that captures the spatial slack and high compression factors in some sensors will be discussed subsequently.

3.2 The Sensor Placement Algorithm

Let us first describe the algorithm for the formulation where we ignore the compression error. Minimizing the cost function in equation 25 is NP-hard in general and therefore we present the following heuristic.

1. Calculate \(w_i = \sum_{\forall y \in y_{grid}} c_{ij}^2\) for all the potential sensor locations (all grids). Here \(c_{ij}\) is the correlation between sensor location \(i\) and grid \(j\).
2. Select the node with largest \(w_i\). This is our first sensor.
3. Now iteratively add sensors to the set of sensors in the following greedy fashion.
3.1 Add a sensor location that results in the maximum decrease in the cost function of equation 25.
3.2 If the total number of sensors is less than \(n\) then repeat else exit.

3.3 Incorporating Fusion Center and Sensor Design Considerations

Once a sensor placement is decided, the techniques presented in the previous section are used to control the granularity of the information transmitted from these sensors to the central register. The finite size \(M\) of this register and the finite wireing space as well as the lack of informativeness of the located sensors forces us to perform this compression. Also sensors may be compressed to a very high degree. This is because while placing sensors we did not account for these physical limitations. The placement algorithm is a heuristic and may not reach the global optimal solution. We would like to use this information pertaining to compression factor of the placed sensors, to replace some sensors with the target of improving the quality of the solution. Basically, we would like to account for the compression error while performing the placement. We present a simple yet effective feedback system by defining a scaling factor \(s_i\) for each placed sensor \(i\) which is proportional to the degree of compression (small \(s_i\) implies high compression). We get rid of all sensors (say \(k\)) whose scaling factor is below a threshold. If doing this removes all the sensors then the threshold is too high. Now, for all the leftover sensors, we modify the correlation \(c_{ij}\) with \(c_{ij} \times s_i\). This implies that due to the compression, the information contained by sensor \(i\) gets reduced as well. We recompute the placement cost using the new correlations in equation 25 (after removing \(k\) sensors). Now we add exactly \(k\) new sensors in the same greedy fashion as described above. We then re-evaluate the compression policy and iterate if necessary.

4. THE COMPLETE DESIGN FLOW

In order to develop this sensing infrastructure, we first generate all the necessary statistical information for the grids and sensors [10]. Then we generate an initial sensor placement while ignoring the compression factors of sensors. Using this sensor placement, we decide the sensor compression factors. The maximum number of bits that a sensor can get (\(b_i\) in equation 10) depends on how much space we have to route its data. In general, sensor locations which are further from the central register and/or in congested areas could be given a smaller \(b_i\). In this way the area and routing overhead can easily be incorporated by appropriate choice of \(M\) and \(b_i\). The compression algorithm distributes the bits to sensors globally and also decides the compression policy locally at each sensor (which observations to compress), thereby giving design specifications to the encoder stage of the sensor. This information is then fed back to the placement engine which accounts for the compression by scaling the correlation information of the sensors appropriately. Then it generates a new placement while accounting for the compression. This process is repeated until the solution converges.

It is noteworthy that our approach accounts for sensor design, compression, fusion and sensor placement in one unified perspective. It also accounts for the space available for placing sensors, central register and routing the data wires.

5. EXPERIMENTAL RESULTS

In this section we present our experimental results. We conducted a set of experiments for both the noiseless and the noisy sensor cases while assuming the sensors can be either infinitely precise (no sensor compression) or are bounded by a total number of 16 bits. We compared the solution (placement, compression and fusion) given by our statistical design framework and that of a range-based method similar to those described in [1, 3] (when the total number of bits are limited we uniformly distribute the bits among all sensors). We used a 16x16 gridding granularity for both methods. For fair comparison, we only calculate and compare the average RMS error for all hotspot locations whose temperature is above a certain threshold (say, 80% of the peak temperature across the chip). This is because most of the range-based methods only allocate sensors to monitor the temperatures at such hot chip areas. If a hotspot is identified and falls within the coverage region of a certain sensor then its temperature will be reported as the sensor obser-
Table 1: RMS error and runtime for different experimental settings

<table>
<thead>
<tr>
<th>#sensors</th>
<th>RMS for range-based method (°C)</th>
<th>RMS for our method (°C)</th>
<th>Improvement (for C/N case)</th>
<th>Runtime (seconds)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>NC/NL</td>
<td>C/NL</td>
<td>C/N</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>11.65</td>
<td>11.69</td>
<td>11.73</td>
<td>11.74</td>
</tr>
<tr>
<td>3</td>
<td>15.45</td>
<td>15.11</td>
<td>10.37</td>
<td>10.49</td>
</tr>
<tr>
<td>4</td>
<td>12.84</td>
<td>14.63</td>
<td>9.08</td>
<td>9.28</td>
</tr>
<tr>
<td>5</td>
<td>11.81</td>
<td>14.49</td>
<td>8.82</td>
<td>8.70</td>
</tr>
<tr>
<td>6</td>
<td>11.34</td>
<td>13.84</td>
<td>7.94</td>
<td>8.66</td>
</tr>
</tbody>
</table>

Figure 5 highlights how the RMS error changes when we increase the central register size constraint M from 8 to 32 while fixing the number of sensors to 6 (with a fixed placement scheme). We can observe a steady decrease in the RMS error for the noiseless case and a first-decrease-then-converge curve for the noisy case. This is because in the noiseless case the sensors can accurately reflect its local temperatures, the more bits we have the more accurate our estimates will be. However in the noisy sensor case, because the sensor observations are corrupted by noise, more bits at each sensor do not necessarily help us gain extra information and hence the error tends to converge after a certain point.

6. CONCLUSION

In this paper we addressed the problems of (1) sensor placement; (2) sensor compression; (3) fusion center design; (4) accounting for the interplay among the above three. We proposed a statistical design framework for solving the whole problem in a unified way. Our results demonstrated that such a statistical design framework can generate an effective thermal sensing infrastructure while meeting various local and global constraints.

7. REFERENCES