A Statistical Methodology for Noise Sensor Placement and Full-Chip Voltage Map Generation

Xiaochen Liu[1], Shupeng Sun[2], Pingqiang Zhou[1], Xin Li[2] and Haifeng Qian[3]
[1]School of Information Science and Technology, ShanghaiTech University, Shanghai, P. R. China
[2]ECE Department, Carnegie Mellon University, Pittsburgh, PA, USA
[3]IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
{liuxch, zhoupq}@shanghaitech.edu.cn; {shupengs, xinli}@ece.cmu.edu; qianhaifeng@us.ibm.com

ABSTRACT

Noise margin violation, also known as voltage emergency induced by continuously reducing noise margin and increasing magnitude of current swings, is becoming a severe threat to the correct execution of applications in processors. Noise sensors can be placed in the non-function area of processors to detect such emergencies by monitoring runtime voltage fluctuations. In this work, we aim to accurately predict the voltage droops using a small set of sensors. We achieve our goal in two steps: We first propose a methodology via group lasso approach to select the optimal set of noise sensors, then build a practical model via ordinary least-squares fitting approach to predict the voltage in the function area of the chip, using the selected sensors in non-function area. Experiment results show that when compared to the full-chip voltage transient simulation, the prediction error of our model is much less than 0.01, and compared to prior work, our approach can achieve better error rates of voltage emergency detection (less than half).

1. INTRODUCTION

Power supply noise has been a long-standing threat to reliable power delivery in processor design, mainly in the form of noise margin violation (also called *voltage emergency*) which happens when voltage fluctuates beyond safe operating margin, causing an increase in the path delay and eventually resulting in the intermittent faults in circuit operation. This problem has gained more and more importance as the CMOS technology continues to scale: on one hand, the mismatch between scaling levels of supply voltage and threshold voltage has gradually reduced the safe operating noise margin [1]; on the other hand, power reduction techniques, such as power gating or clock gating, lead to large current swings over a relatively small time scale when they are applied in microprocessors to throttle power consumption.

Recently several researchers and processor vendors have proposed various approaches to handle this problem. Some of them [2–5] allow voltage emergencies to occur. When emergencies are detected, recovery mechanisms will be applied to roll back the architectural state, such as the registers and memory state, to a guaranteed-correct state. Because such rollback mechanisms can be prohibitively expensive, researchers have also developed voltage emergency predictors [6–9], based on current/voltage profiles, microarchitectural signatures etc., to identify impending emergencies and prevent their occurrence by throttling mechanisms. These throttling mechanisms try to reduce current variations by either reducing the processor's clock rate [10], or activating idle functional units when there is a sudden reduction in current draw [6], or controlling instruction issue and current change in the pipelines [11, 12].

All these emergency-handling mechanisms rely on sensors to detect impending or occurred voltage or current margin violations. For example, in Intel processors, error detection circuits EDS and TRC [2] are used to achieve delay fault detection; in IBM POWER7 server [7], a sensor-based throttling approach called Critical Path Monitoring (CPM) has been implemented to measure the available timing margin in real time.

Our work is motivated by the following observations when sensors are used to detect voltage emergencies on chip:

- The number of sensors that can be used in a chip is limited due to the large hardware design overhead associated with the application of voltage sensors. Therefore, to build a practical noise monitoring system, we should optimize the placement of the voltage sensors.
- The chip area where we can put voltage sensors are restricted and they may not reside the function area, so there exists mismatch between *the voltage measured by the voltage sensors* and *the voltage we would like to monitor in the function area*. Such mismatch can potentially incur two types of errors: *miss error*, that happens when there is no emergency in the locations we put sensors, but the emergencies do occur in the function area, and *wrong alarm error*, that happens when the sensors alarm emergencies, but there is no emergency in the monitored function area. To eliminate these two types of errors, we need to build an accurate model to predict the voltage in the function area, based on the voltage we obtain from the placed sensors. Luckily this prediction is feasible because the noise in the local area of a power grid is highly correlated [13].

There has been very limited prior work in the literature to solve the voltage sensor placement problem and to predict the voltage in the function area. Wang *et al.* [13] proposed a statistical framework, named Eagle-Eye, for voltage sensor placement, with the objective of minimizing *miss error* only. In our work, we propose a practical methodology to 1) solve the sensor placement problem, and 2) to build the voltage map of the full-chip area based on the measured voltage by placed sensors. We exploit the strong correlations among the voltage noises of nodes in a local power grid and build a model that captures the relationship between *the voltages at the sensor candidate locations* and *the voltages at the noise critical locations (whose voltages we would like to monitor) in*

DAC '15, June 07-11, 2015, San Francisco, CA, USA
Copyright 2015 ACM 978-1-4503-3520-1/15/06 ...$15.00
http://dx.doi.org/10.1145/2744769.2744784.

the function area. For building this model, we apply Group-Lasso (GL) to fit the model coefficients, which are then used to select the most important sensors out of the given sensor candidate set. After solving the sensor placement problem, we further build an accurate prediction model between the *voltages measured in the placed sensors* and *the voltages at the noise critical locations in the function area*, via ordinary least squares (OLS) approach, to reduce both the miss errors and wrong alarm errors.

The contributions of this paper are summarized below:

- we propose a novel method for sensor placement based on GL, and then build an accurate model to predict the voltage in the function area, using the voltages measured by the placed sensors. Experiment results show that the prediction accuracy of our model is high even with a small number of placed sensors on chip.
- we compare our approach with prior work. Experiment results show that our approach is superior in reducing the emergency detection errors in the function area, based on the information from the placed sensors in non-function area.

The rest of the paper is organized as follows. In Section 2.1 we present the overview of our methodology. We describe the approach to solve the sensor placement problem based on GL in Section 2.2, and then present the OLS approach to build an accurate prediction model in Section 2.3. The effectiveness of our proposed methodology is verified by the experimental results given in Section 3. Finally, conclusions are made in Section 4.

2. OUR METHODOLOGY

2.1 Overview

Without loss of generality, we partition the chip into two areas: function area (FA) and blank area (BA). Here, FA refers to the area where we put the circuit blocks, and BA refers to the remaining area on the chip. Though both the supply voltages in FA and BA may have large variations, we are more interested in the supply voltages in FA since they directly affect the performance of the system. One simple way to monitor the supply voltages in FA is to put sensors inside it. If the sensor detects a voltage emergency, we know that the corresponding circuit block in FA may malfunction. Such idea, however, is extremely difficult to achieve due to the fact that FA may not have enough space to put voltage sensors.

Realizing this limitation, we propose a novel model-based approach to accurately monitor the supply voltages in FA. Our proposed approach is based on two important observations. First, though FA may not have enough space for sensor allocation, we can put sensors in BA to obtain its supply voltages. Second, the supply voltages in FA and BA can be strongly correlated, which has been well known by both industry and academia [13]. Having these observations, the key idea of our proposed approach is to predict the supply voltages in FA from the measured voltages in BA.

To this end, an important task is to build models to accurately capture the mathematical mapping from the voltages in BA to the voltages in FA. Without losing generality, we suppose that there are M nodes in BA where we can put voltage sensors, and the M-dimensional vector

$$\boldsymbol{x} = [x_1 \quad x_2 \quad \dots \quad x_M]^T \tag{1}$$

contains the voltages of these M sensor candidate locations, where x_m denotes the voltage at the m-th sensor candidate location. We further suppose that there are K functional circuit blocks in FA, and the K-dimensional vector

$$\boldsymbol{f} = [f_1 \quad f_2 \quad \dots \quad f_K]^T \tag{2}$$

contains the supply voltages of these K circuit blocks, where f_k denotes the worst supply voltage at the k-th circuit block. Note that in our current work, we select one representative node for each circuit block. However, it is easy for our model to handle the case with more representative nodes per block. Due to the fact that $\boldsymbol{x}$ and $\boldsymbol{f}$ can be modeled as random variables and they are strongly correlated, we can approximate each of $\{f_k;\ k = 1, 2, \dots, K\}$ as a linear function of $\boldsymbol{x}$:

$$\left.\begin{aligned} f_1 &\approx \sum_{m=1}^{M} \alpha_{1,m} \cdot x_m + c_1 \\ f_2 &\approx \sum_{m=1}^{M} \alpha_{2,m} \cdot x_m + c_2 \\ &\vdots \\ f_K &\approx \sum_{m=1}^{M} \alpha_{K,m} \cdot x_m + c_K \end{aligned}\right. \quad \Rightarrow \boldsymbol{f} \approx \boldsymbol{\alpha} \cdot \boldsymbol{x} + \boldsymbol{c} \tag{3}$$

where

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots & \alpha_{1,M} \\ \alpha_{2,1} & \alpha_{2,2} & \dots & \alpha_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{K,1} & \alpha_{K,2} & \dots & \alpha_{K,M} \end{bmatrix} \tag{4}$$

denotes the model coefficients, and

$$\boldsymbol{c} = [c_1 \quad c_2 \quad \dots \quad c_K]^T \tag{5}$$

denotes the constant terms.

In Eq. (3), the supply voltage in each circuit block k is predicted from the voltages at all the M sensor candidate locations. Though the prediction error could be pretty small, the design overhead for deploying all the M candidate sensors could be unacceptable. For instance, the power consumed by these M sensors can be huge, which makes the proposed methodology unattractive. To address this issue, we aim to find a small number of (say, Q) sensors from all the M candidates, and accurately predict the supply voltages in FA based on these Q selected sensors. Towards this goal, we need to answer two fundamental questions:

1. how to select these Q sensors, and
2. how to learn the model coefficients in Eq. (3) after sensor selection.

In what follows, we will first introduce the group lasso technique to help us select the most important sensors in Section 2.2, and then discuss the mathematical details of model construction via ordinary least squares fitting in Section 2.3. Finally, we summarize our methodology in Section 2.4.

2.2 Sensor Placement via Group Lasso

In this section, we aim to select a small number of (say, Q) important sensors from all the M sensor candidates via group lasso (GL) technique so that the supply voltages in FA can be accurately predicted based on these Q selected sensors. Intuitively speaking, if all the model coefficients in a column of $\boldsymbol{\alpha}$ are close to zero, it implies that the sensor associated with this column has negligible contributions to accurate model prediction and, hence, should not be selected. From this point of view, selecting a small number of important sensors is equivalent to finding a sparse solution of $\boldsymbol{\alpha}$ where most columns have close-to-zero model coefficients.

Without loss of generality, we assume that N sampling data points are first collected

$$\boldsymbol{X} = [\boldsymbol{x}^{(1)}\ \boldsymbol{x}^{(2)}\ \dots\ \boldsymbol{x}^{(N)}] = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(N)} \\ x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(N)} \\ \vdots & \vdots & \ddots & \vdots \\ x_M^{(1)} & x_M^{(2)} & \dots & x_M^{(N)} \end{bmatrix}$$
$$\boldsymbol{F} = [\boldsymbol{f}^{(1)}\ \boldsymbol{f}^{(2)}\ \dots\ \boldsymbol{f}^{(N)}] = \begin{bmatrix} f_1^{(1)} & f_1^{(2)} & \dots & f_1^{(N)} \\ f_2^{(1)} & f_2^{(2)} & \dots & f_2^{(N)} \\ \vdots & \vdots & \ddots & \vdots \\ f_K^{(1)} & f_K^{(2)} & \dots & f_K^{(N)} \end{bmatrix} \tag{6}$$

where $\boldsymbol{x}^{(n)}$ and $\boldsymbol{f}^{(n)}$ denote the sensor voltages $\boldsymbol{x}$ and supply voltages $\boldsymbol{f}$ for the n-th data point respectively. A simple way to solve the model coefficients $\boldsymbol{\alpha}$ in (4) is to apply the traditional ordinary least squares (OLS) fitting approach [14]:

$$\min_{\boldsymbol{\alpha},\boldsymbol{c}} \|\boldsymbol{F} - \boldsymbol{\alpha}\cdot\boldsymbol{X} - \boldsymbol{C}\|_F \tag{7}$$

where

$$\boldsymbol{C} = [\boldsymbol{c} \quad \boldsymbol{c} \quad \dots \quad \boldsymbol{c}]_{K\times N} \tag{8}$$

and $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, i.e., the square root of the summation of the absolute squares of all the elements in the matrix. Intuitively, OLS intends to find a solution that can minimize the mean squared modeling error.

As mentioned at the beginning of this section, we aim to find a sparse solution of $\boldsymbol{\alpha}$ where most columns have close-to-zero model coefficients. However, the OLS formulation in (7) poses no constraint on the sparsity of $\boldsymbol{\alpha}$. To select important sensors, one intuitive idea is to select the sensors with large components in $\boldsymbol{\alpha}$, assuming that the importance of a sensor is directly correlated to the value of its corresponding elements in $\boldsymbol{\alpha}$. Unfortunately, this idea may not always work because of the complexity in feature selection [15]. Therefore, the unconstrained optimization in (7) used by OLS cannot fit our need of sensor selection.

Realizing this limitation of OLS, we adopt group lasso (GL) technique from the statistics community to help us select a small number of important sensors [16,17]. To apply GL, $\boldsymbol{x}$ and $\boldsymbol{f}$ need to be normalized to have zero mean and unit variance. Denote $\boldsymbol{z}$ and $\boldsymbol{g}$ as the normalized $\boldsymbol{x}$ and $\boldsymbol{f}$ respectively. Eq. (3) can be rewritten as

$$\begin{aligned} g_1 &\approx \sum_{m=1}^{M} \beta_{1,m}\cdot z_m \\ g_2 &\approx \sum_{m=1}^{M} \beta_{2,m}\cdot z_m \\ &\ \ \vdots \\ g_K &\approx \sum_{m=1}^{M} \beta_{K,m}\cdot z_m \end{aligned} \quad \Rightarrow \boldsymbol{g} \approx \boldsymbol{\beta}\cdot\boldsymbol{z} \tag{9}$$

where

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_{1,1} & \beta_{1,2} & \dots & \beta_{1,M} \\ \beta_{2,1} & \beta_{2,2} & \dots & \beta_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ \beta_{K,1} & \beta_{K,2} & \dots & \beta_{K,M} \end{bmatrix} \tag{10}$$

denotes the model coefficients of the linear models shown in (9). After normalization of $\boldsymbol{X}$ and $\boldsymbol{F}$, we have

$$\begin{aligned} \boldsymbol{Z} &= [\boldsymbol{z}^{(1)}\ \boldsymbol{z}^{(2)}\ \dots\ \boldsymbol{z}^{(N)}] \\ \boldsymbol{G} &= [\boldsymbol{g}^{(1)}\ \boldsymbol{g}^{(2)}\ \dots\ \boldsymbol{g}^{(N)}] \end{aligned} \tag{11}$$

where $\boldsymbol{z}^{(n)}$ and $\boldsymbol{g}^{(n)}$ denote the normalized $\boldsymbol{x}^{(n)}$ and $\boldsymbol{f}^{(n)}$ in (6) respectively. Based on (9) and (11), GL formulates the following optimization problem

$$\begin{aligned} &\min_{\boldsymbol{\beta}} \|\boldsymbol{G} - \boldsymbol{\beta}\cdot\boldsymbol{Z}\|_F \\ &s.t. \quad \|\boldsymbol{\beta}_1\|_2 + \|\boldsymbol{\beta}_2\|_2 + \dots + \|\boldsymbol{\beta}_M\|_2 \le \lambda \end{aligned} \tag{12}$$

where $\boldsymbol{\beta_m}$ denotes the m-th column in $\boldsymbol{\beta}$, $\|\cdot\|_2$ denotes the l_2-norm of a vector and λ is a user-defined hyper-parameter.

From (12), we have several important observations. First, $\boldsymbol{\beta_m}$ where $m \in \{1, 2, \dots, M\}$, includes all the model coefficients related with the m-th sensor. If the p-th sensor is more important than the q-th sensor, $\|\boldsymbol{\beta}_p\|_2$ should be larger than $\|\boldsymbol{\beta}_q\|_2$, where $p, q \in \{1, 2, \dots, M\}$. When λ is sufficiently small, it is very likely that $\|\boldsymbol{\beta}_q\|_2$ is close to zero given the constraint in (12). In other words, the unimportant sensors are likely to have extremely small model coefficients after solving (12). Hence, we can identify the set of important sensors based on the values of $\|\boldsymbol{\beta}_m\|_2$. To this end, we can define a threshold value (say, T) which should be a very small value. If $\|\boldsymbol{\beta}_m\|_2 > T$, the m-th sensor is considered as important. Otherwise, the m-th sensor is not selected.

Another important observation is that the hyper-parameter λ plays an extremely important role in sensor selection. Intuitively speaking, if λ is small, most of $\|\boldsymbol{\beta}_m\|_2$ are close to zero and, therefore, a very small number of sensors are considered as important. Due to the page limit, detailed discussions about the relation between the value of λ and the number of selected sensors are omitted in this paper, but can be found in the literature [16]. To reduce the design overhead, we should set λ to be a relatively small value. However, if the number of selected sensors is too small, the linear models in Eq. (3) may not be sufficiently accurate. How to determine the value of λ depends both on the design overhead that we can afford and the prediction accuracy that we require. In Section 2.4, we will discuss how to choose an appropriate value of λ.

Eq. (12) can be re-formulated as a second-order cone programming problem, and then efficiently solved by interior point method [18]. Once (12) is solved, we can identify a small number of (say, Q) important sensors based on the values of $\|\boldsymbol{\beta}_m\|_2$, as previously discussed. Here, we denote the indexes of Q selected sensor as

$$\boldsymbol{S} = \{S_1, S_2, \dots, S_Q\} \tag{13}$$

where S_q represents the index of the q^{th} selected sensor. For instance, assume that we have 100 (i.e., $M = 100$) sensor candidates in total, and the 2^{nd} sensor and the 17^{th} sensor are identified as important sensors. Then $\boldsymbol{S} = \{2, 17\}$. In the next section, we will discuss how to accurately predict the supply voltages $\boldsymbol{f}$ in FA via these Q sensors.

2.3 Prediction Model Construction via OLS Fitting

A straightforward way to predict the supply voltages in FA is to apply the linear models learned from (12). Namely,

$$g_k^* = \beta_{k,S_1}\cdot z_{S_1} + \beta_{k,S_2}\cdot z_{S_2} + \dots + \beta_{k,S_Q}\cdot z_{S_Q} \tag{14}$$

where $k = 1, 2, \dots, K$, $q = 1, 2, \dots, Q$, β_{k,S_q} denotes the model coefficient learned from the GL optimization in (12), z_{S_q} denotes the normalized measured voltage of the q-th selected sensor, and g_k^* denotes the predicted value of g_k. Once g_k^* is available, we can recover f_k by applying inverse normalization.

The linear models in (14), however, may not result in an accurate prediction. To clearly understand this, let us consider a simple example where we have two sensor candidates (i.e., $\boldsymbol{z} = [z_1\ z_2]^T$) and two circuit blocks (i.e., $\boldsymbol{g} = [g_1\ g_2]^T$).

For illustration purposes, we assume that

$$g_1 = g_2 = z_1 \tag{15}$$

As mentioned in Section 2.2, to select a small number of important sensors, λ should be set to a relatively small value. In this simple example, we assume λ is set to 1, and it is easy to see that only the first sensor will be selected after solving the GL optimization problem in (12). Even though we successfully select the most important sensor in this simple example, the model coefficients $\{\beta_{k,1};\ k = 1, 2\}$ solved from (12) can be quite different from their optimal value {1,1} since $\{\beta_{k,1}; k = 1, 2\}$ need to satisfy the following constraint

$$\sqrt{\beta_{1,1}^2 + \beta_{2,1}^2} \leq \lambda = 1. \tag{16}$$

Intuitively speaking, the model coefficients solved from (12) for the selected sensors can be highly biased due to the constraint posed on the model coefficients. As such, the linear models in (14) do not fit our need of accurate model prediction.

To address this issue, we apply OLS to solve the following unconstrained optimization problem

$$\min_{\boldsymbol{\alpha}^S, \boldsymbol{c}} \|\boldsymbol{F} - \boldsymbol{\alpha}^S \cdot \boldsymbol{X}^S - \boldsymbol{C}\|_F \tag{17}$$

where

$$\boldsymbol{X}^S = \begin{bmatrix} x_{S_1}^{(1)} & x_{S_1}^{(2)} & \cdots & x_{S_1}^{(N)} \\ x_{S_2}^{(1)} & x_{S_2}^{(2)} & \cdots & x_{S_2}^{(N)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{S_Q}^{(1)} & x_{S_Q}^{(2)} & \cdots & x_{S_Q}^{(N)} \end{bmatrix} \tag{18}$$

$$\boldsymbol{\alpha}^S = \begin{bmatrix} \alpha_{1,S_1} & \alpha_{1,S_2} & \cdots & \alpha_{1,S_Q} \\ \alpha_{2,S_1} & \alpha_{2,S_2} & \cdots & \alpha_{2,S_Q} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{K,S_1} & \alpha_{K,S_2} & \cdots & \alpha_{K,S_Q} \end{bmatrix}, \tag{19}$$

$\boldsymbol{F}$ is defined in (6), and $\boldsymbol{C}$ is defined in (8). Once $\boldsymbol{\alpha}^S$ and $\boldsymbol{c}$ are determined, we can predict the supply voltages in FA based on the following linear models

$$f_k^* = \alpha_{k,S_1} \cdot x_{S_1} + \alpha_{k,S_2} \cdot x_{S_2} + \cdots + \alpha_{k,S_Q} \cdot x_{S_Q} \tag{20}$$

where α_{k,S_q} $(k = 1, 2, \ldots, K,\ q = 1, 2, \ldots, Q)$ denotes the model coefficient learned from the OLS optimization in (17), $\{x_{S_q}\}$ $(q = 1, 2, \ldots, Q)$ denote the measured voltages of Q selected sensors, and f_k^* $(k \in \{1, 2, \ldots, K\})$ denotes the predicted value of f_k.

2.4 Summary

Our proposed sensor selection and model construction technique can be summarized as follows:

- Step 0: Start from a pre-defined λ, a threshold value T, M sensor candidates, and K circuit blocks.
- Step 1: Collect N sampling data points $\{\boldsymbol{x}^{(n)}\}$ and $\{\boldsymbol{f}^{(n)}\}$ where $n = 1, 2, \ldots, N$.
- Step 2: Form $\boldsymbol{X} = [\boldsymbol{x}^{(1)} \cdots \boldsymbol{x}^{(N)}]$ and $\boldsymbol{F} = [\boldsymbol{f}^{(1)} \cdots \boldsymbol{f}^{(N)}]$ in (6).
- Step 3: Form $\boldsymbol{Z}$ and $\boldsymbol{G}$ in (11) by normalizing $\boldsymbol{X}$ and $\boldsymbol{F}$.
- Step 4: Solve the GL optimization problem in (12), and then calculate $\{\|\boldsymbol{\beta}_m\|_2; m = 1, 2, \ldots, M\}$.
- Step 5: If $\|\boldsymbol{\beta}_m\|_2 > T$ where $m \in \{1, 2, \cdots, M\}$, the m-th sensor is selected. Otherwise, the m-th sensor is not selected. Assume Q sensors are selected, and their indexes are $\{S_q; q = 1, 2, \cdots, Q\}$.
- Step 6: Form $\boldsymbol{X}^S$ in (18) based on $\boldsymbol{X}$ and $\{S_q; q = 1, 2, \cdots, Q\}$.
- Step 7: Learn the model coefficients $\boldsymbol{\alpha}^S$ in (19) by solving the OLS optimization problem in (17).
- Step 8: Form the prediction models in (20) using $\boldsymbol{\alpha}^S$.

There are two clarifications we need to make for the above steps. First, different λ values result in different number of (i.e., Q) selected sensors. To choose an appropriate λ so that a small number of sensors are selected and the prediction models in (20) are satisfactorily accurate, we typically sweep the value of λ in a large range. Namely we start from a small λ, run steps 1–8, resulting in very compact prediction models in (20). Next, we increase λ, and repeat Steps 4–8. The prediction model will become more accurate at the expense of utilizing more sensors. The aforementioned flow proceeds until λ reaches the maximal value in the sweeping range. As long as the sweeping range is large enough, we should be able to find an appropriate λ so that the prediction models in (20) are sufficiently accurate by using a relatively small number of sensors. Second, the values of $\|\boldsymbol{\beta}_m\|_2$ for selected sensors are much larger than those for un-selected sensors, which is demonstrated by our experimental results in Section 3.1. Hence, it is easy to determine the threshold value T so that important sensors and unimportant sensors are appropriately classified.

Note that although we perform steps 0 – 8 to generate the prediction model at design time, we only need to evaluate the prediction model in (20) for dynamic noise management at runtime, which is computationally cheap due to the fact that only a small set of sensors are used in most chips.

3. EXPERIMENTAL RESULTS

In this section, we introduce how to obtain the voltage samples of the sensor candidate locations in BA and the noise critical nodes in FA of a given chip, which are used to train our sensor placement approach (see Section 2.2) and voltage prediction model (see Section 2.3).

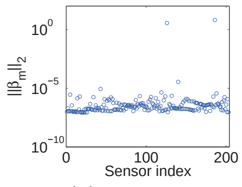
In our experiments, we consider a 22nm homogenous 8-core Intel Xeon E5-like multiprocessor (2.5GHz) with 30 function blocks in each core. We divide the whole chip into two areas: FA, that is the area covered by all the function blocks, and BA, the remaining area in the chip. We select one noise critical node within each function block which has the worst noise during a sampling simulation period, and assume all the nodes in the BA to be candidate nodes for sensors. The following steps are then used to obtain the voltage samples for the noise critical nodes and sensor candidate nodes:

1. We use a full system multicore simulator GEM5 [19] to run the PARSEC 2.1 benchmarks [20] to generate the runtime statistics of all the function blocks.
2. We then use McPAT [21] with power gating enabled to analyze these runtime statistics and generate the power profiles of the function blocks.
3. After that, we perform transient simulation of the power grid for the whole chip, to find the voltage traces of all the power grid nodes, including those noise critical nodes and sensor candidate nodes. In our experiments, the supply voltage VDD is set to be 1.0V.
4. At a certain sampling time point, the voltages of all the nodes on chip form a complete voltage map. So we can generate the full-chip voltage map at a certain time point from the voltage traces of the power grid nodes. In our experiments, we randomly select 10,000 voltage maps out of 19 benchmarks as our training samples.

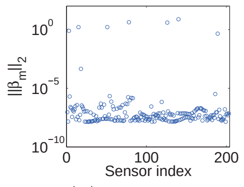
3.1 Sensor selection and model accuracy

In this section, we present the results for our methodology to sensor placement in BA and voltage prediction in FA.

As stated in Section 2.2, the user-defined hyper-parameter λ places a direct constraint on the coefficient of each sensor candidate, $\|\boldsymbol{\beta}_m\|_2$, and further affects the number of sensors we can select for a given chip. Fig. 1 shows the values of



(a) $\lambda = 10$



(b) $\lambda = 30$

Figure 1: $\|\boldsymbol{\beta}_m\|_2$ for sensor candidates in one core.

coefficient $\|\boldsymbol{\beta}_m\|_2$ for each sensor candidate m when $\lambda = 10$ and $\lambda = 30$. Obviously, most $\|\boldsymbol{\beta}_m\|_2$s are around 10^{-5} to 10^{-10}. Therefore, we use a pre-defined threshold T to decide whether a sensor candidate can be selected in the placement: If $\|\boldsymbol{\beta}_m\|_2 > T$, we consider the voltage of candidate m has a strong impact on the prediction of the voltage at the noise critical nodes, thus candidate m is selected; otherwise, candidate m is excluded in the placement. In our experiments, T is set to be 10^{-3}.

Table 1: λ vs. the number of sensors in each core and the aggregated relative prediction error

λ	10	20	30	40	50	60
# of selected sensors	2	4	7	10	13	16
relative error(%)	0.51	0.25	0.11	0.06	0.05	0.04

Table 1 clearly shows that the number of chosen sensors for one core in the placement increases as λ goes up. This is because a larger λ gives a looser constraint on $\|\boldsymbol{\beta}_m\|_2$, and therefore those candidates that are slightly less important are also included in the placement. After selecting the voltage sensors, we predict the voltages of the noise critical nodes in the FA using our model presented in Section 2.3. Fig. 2 shows part of the voltage traces at one noise critical node respectively predicted by our model with two and seven selected sensors per core. We can see that compared to the real voltage trace obtained by transient simulation, the error of the predicted voltage is quite small, and the prediction error can be further reduced by increasing the number of selected sensors in the model.

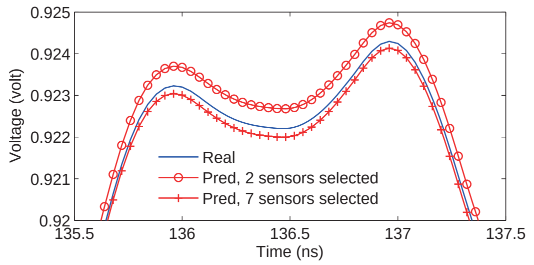


Figure 2: Predicted voltage vs. real voltage of one noise critical node in one core.

Table 1 also shows how the aggregated relative prediction error (for all function blocks and all benchmarks) changes as we increase the parameter λ. As we increase λ from 10 to 40, the relative error decreases very fast, which means that increasing the number of sensors used in the chip can significantly increase the accuracy of our prediction model.

To achieve better prediction results, we should use more sensors for a chip. However, if we consider the area and manufacturing cost of the sensors, we should control the number of used sensors. Luckily, our results show that even when $\lambda = 10$, which means there are only two or three sensors in one core, the relative prediction error is lower than 10^{-2}. In the real design, the designer can use the parameter λ to explore the tradeoff between the chip design cost and the voltage prediction performance.

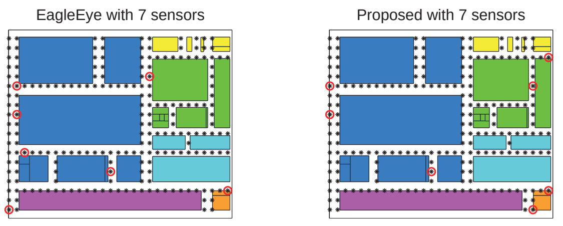


Figure 3: Selected sensors by Eagle-eye and the proposed approach.

Fig. 3 shows the locations of sensors respectively selected by our approach and Eagle-Eye when only seven sensors are available. Blocks that are functionally relative or similar are grouped into one unit as denoted by one color. We can see that Eagle-eye places six of seven sensors around or in the blue-colored unit (the execution unit) because it tends to select the sensor candidates with worst voltage noise. In contrast, our approach selects only four sensors within the blue-colored unit, and spare three sensors for other function units. This is because our approach seeks to find the sensor candidates that have the strongest correlation with the voltages at the circuit blocks, and such candidates may not necessarily have the worst voltage noise.

3.2 Error rate analysis

In this section, we present results to show the effectiveness of our methodology to maximizing the effect of voltage emergency detection. In particular, we compare our methodology with one recent work Eagle-Eye [13], in terms of three error rates related to emergency detection:

- *Miss error (ME) rate*: This is the probability that the emergencies occurred in the FA are not detected by the voltage sensors in Eagle-Eye, or the prediction model in our work.
- *Wrong alarm error (WAE) rate*: This is the probability that there are no emergencies occurred in the FA, but the voltage sensors alarm as in Eagle-Eye, or the prediction model alarms in our work.
- *Total error (TE) rate*: This is the probability that the voltage sensors in Eagle-Eye or the prediction models in our work report a wrong state of the noise critical nodes in the FA. It equals to dividing *the number of samples in which wrong states reported* by *the number of total samples.*

Our work differs with Eagle-Eye in that we use a different approach to select the voltage sensors, and we further build a prediction model based on these selected sensors for the noise critical nodes in the FA. For fair comparison, we use the same benchmarks/voltage samples and chip layout for Eagle-Eye and our methodology. In the experiments, we assume voltage emergency occurs when the supply voltage falls below 0.85V, when VDD is set to be 1.0V.

We compare these two approaches on 19 benchmarks. Due to limited space, we only show part of the results. Compared to Eagle-Eye, our approach can reduce both the miss error rate and total error rate by about half for all the benchmarks (see Table. 2). As for the wrong alarm error rate, when the total number of allocated sensors for the whole chip is large (more than 50), our approach is always superior to Eagle-Eye. However, when the total number of allocated

Table 2: The error results with 2 sensors per core

Benchmark	Eagle-Eye			Proposed		
	ME	WAE	TE	ME	WAE	TE
BM1	0.0976	0.0003	0.0330	0.0420	0.0006	0.0145
BM2	0.0822	0.0004	0.0270	0.0351	0.0004	0.0116
BM3	0.1152	0.0001	0.0208	0.0507	0.0007	0.0097
BM4	0.0889	0.0002	0.0307	0.0368	0.0001	0.0127
BM5	0.0976	0.0001	0.0386	0.0385	0.0002	0.0153
BM6	0.2162	0.0001	0.0386	0.0385	0.0002	0.0153
BM7	0.1431	0.0007	0.0362	0.0659	0.0003	0.0167
BM8	0.1130	0.0003	0.0287	0.0494	0.0002	0.0126
BM9	0.0888	0.0003	0.0294	0.0362	0.0002	0.0120
BM10	0.1256	0.0001	0.0281	0.0565	0.0001	0.0126
BM11	0.1256	0	0.0301	0.0490	0.0006	0.0122
BM12	0.0612	0.0003	0.0246	0.0270	0.0001	0.0109
BM13	0.1159	0.0001	0.0362	0.0410	0	0.0128
BM14	0.0905	0.0009	0.0332	0.0425	0.0001	0.0154
BM15	0.1118	0.0002	0.0384	0.0400	0.0005	0.0140
BM16	0.0977	0.0007	0.0267	0.0436	0.0006	0.0121
BM17	0.1231	0.0002	0.0434	0.0489	0.0002	0.0173
BM18	0.1395	0	0.0400	0.0506	0.0001	0.0145
BM19	0.1289	0.0002	0.0437	0.0493	0.0001	0.0167

sensors is small (for example, less than 30), our approach achieves better results on 11 benchmarks (see Fig. 4 for one such case), while Eagle-Eye does a little better on the other 8 benchmarks. This is because 1) in sensor placement, Eagle-Eye tends to select the candidate locations with worst voltage noise, and therefore have less probability to alarm wrong emergencies; our approach prefers to select candidate locations whose voltages have strong correlation with the voltages at the noise critical nodes, but such candidate locations may not have worst voltage noise, 2) the prediction accuracy of our model is not high enough when the number of allocated sensors is small. Fortunately, as we can from Table. 2, wrong alarm error rate is generally small (less than 10^{-3}), and miss error rate dominates the total error rate. Furthermore, in our work, we assume that the sensors can only be placed in blank area. In fact, it is possible for the designers to place the sensors inside the function area, to further improve the prediction accuracy of our model and therefore achieve smaller error rates.

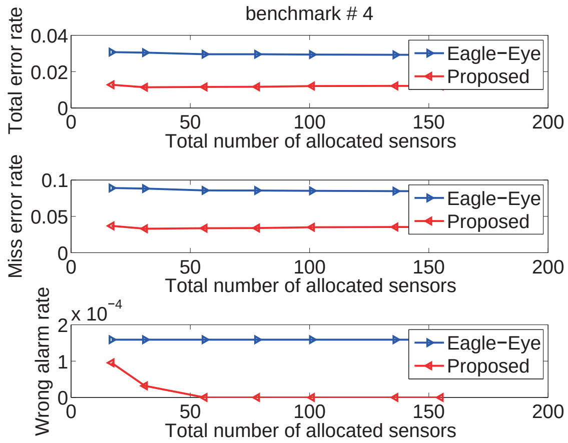


Figure 4: The error rate results of BM4.

4. CONCLUSION

We propose a novel methodology to allocate the noise sensors in a given chip via group lasso approach, and then build an ideal model via ordinary least-squares fitting approach to predict the voltage in the function area of the chip, using the voltages measured by the placed sensors in blank area. Results show that compared to prior work, our approach can achieve superior results in terms of the emergency detection errors, and the prediction error of our model can be pretty high (less than 10^{-2}) even with a small set of sensors.

5. REFERENCES

[1] ITRS report, http://www.itrs.net/.

[2] J. Tschanz *et al.*, "A 45nm resilient and adaptive microprocessor core for dynamic variation tolerance," in *ISSCC*, 2010, pp. 282–284.

[3] M. Gupta *et al.*, "DeCoR: a delayed commit and rollback mechanism for handling inductive noise in microprocessors," in *HPCA*, 2008, pp. 381–392.

[4] M. Mack *et al.*, "IBM POWER6 reliability," *IBM Journal of Research and Development*, vol. 51, no. 6, pp. 763–774, Nov. 2007.

[5] D. Ernst *et al.*, "Razor: A low-power pipeline based on circuit-level timing speculation," in *MICRO*, 2003, pp. 7–18.

[6] R. Joseph *et al.*, "Control techniques to eliminate voltage emergencies in high-performance processors," in *HPCA*, 2003, pp. 79–90.

[7] C. R. Lefurgy *et al.*, "Active guardband management in power7+ to save energy and maintain reliability," *IEEE Micro*, vol. 33, no. 4, pp. 35–45, Jul. - Aug. 2013.

[8] M. Gupta *et al.*, "An event-guided approach to reducing voltage noise in processors," in *DATE*, 2009, pp. 160–165.

[9] V. Reddi *et al.*, "Voltage emergency prediction: using signature to reduce operating margins," in *HPCA*, 2009, pp. 18–29.

[10] E. Grochowski *et al.*, "Microarchitectural simulation and control of di/dt-induced power supply voltage variation," in *HPCA*, 2002, pp. 7–16.

[11] M. Powell and T. Vijaykumar, "Pipeline muffling and a priori current ramping: architectural techniques to reduce high-frequency inductive noise," in *ISLPED*, 2003, pp. 223–228.

[12] M. Powell and T. Vijaykumar, "Pipeline damping: a microarchitectural technique to reduce inductive noise in supply voltage," in *ISCA*, 2003, pp. 72–83.

[13] T. Wang *et al.*, "Eagle-Eye: a near-optimal statistical framework for noise sensor placement," in *ICCAD*, 2013, pp. 437–443.

[14] C. Bishop, *Pattern Recognition and Machine Learning.* Prentice Hall, 2007.

[15] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[16] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of Royal Statistical Society*, vol. 68, no. 1, pp. 49–67, February 2006.

[17] C. Hsu *et al.*, "Test data analytics – exploring spatial and test-item correlations in production test data," in *ITC*, 2013, pp. 1–10.

[18] M. Lobo *et al.*, "Applications of second-order cone programming," *Linear Algebra and Its Applications*, vol. 284, pp. 193–228, Nov. 1998.

[19] N. Binkert *et al.*, "The gem5 simulator," in *ACM SIGARCH Computer Architecture News*, May 2011, pp. 1–7.

[20] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, January 2011.

[21] S. Li *et al.*, "McPAT: an integrated power, area, and timing modeling framework for multicore and manycore architectures," in *MICRO*, 2009, pp. 469–480.