A Fast Leakage-Aware Full-Chip Transient Thermal Estimation Method

Hai Wang, Jiachun Wan, Sheldon X.-D. Tan, *Senior Member, IEEE*, Chi Zhang, He Tang, *Member, IEEE*, Yuan Yuan, Keheng Huang, and Zhenghong Zhang

Abstract—Accurate and fast thermal estimation is important for the runtime thermal regulation of modern microprocessors due to excessive on-chip temperatures. However, due to the nonlinear relationship between the leakage power and temperature, full-chip thermal estimation methods suffer slow speed and scalability issue when the increasing static leakage power is considered. In this work, we propose a new fast leakage-aware full-chip thermal estimation method. Unlike traditional methods, which use iteration to handle the leakage-temperature nonlinearity dependency issue, the new method applies a dynamic linearization algorithm, which adaptively transforms the original nonlinear thermal model into a number of local linear thermal models. In order to further improve the thermal estimation efficiency, a specially-designed adaptive model order reduction method is integrated into the thermal estimation framework to generate local compact thermal models. Our numerical results show that the new method is able to accurately estimate full-chip transient temperature distribution by fully considering the nonlinear leakage-temperature dependency with fast speed. On different chips with core number ranging from 9 to 36, it achieved 85$\times$ to 589$\times$ speedup in average against traditional iteration based method, with average thermal estimation error to be around 0.2°C.

Index Terms—Thermal estimation, transient analysis, leakage, full-chip

1 Introduction

Thermal and its related reliability issues have become the primary concerns for high performance microprocessors, especially after the breakdown of the so-called Dennard scaling, since power density starts to increase as IC technology advances [1], [2]. To enhance reliability, researchers have proposed many thermal regulation or dynamic thermal management methods, including clock gating, power gating, Dynamic Voltage and Frequency Scaling (DVFS), and task migration techniques [3], [4], [5], [6].

To make all of those on-chip thermal management techniques work, one critical aspect is to correctly estimate the full-chip temperature profile. Some existing methods rely on on-chip physical thermal sensors to make thermal regulation decisions. However, the limitation in those methods is that only very few physical thermal sensors are available, thus the temperature information obtained only from sensors may be insufficient or sometimes misleading for thermal regulation decision making. On the other hand, obtaining on-chip temperature information by runtime full-chip thermal estimation becomes a more practical solution. These methods first construct the thermal model of the processor, and then calculate the thermal estimation based on the power estimation as inputs to the thermal model [7], [8]. As a result, they are able to obtain temperatures at positions where there are no physical thermal sensors.

One major drawback of existing runtime full-chip estimation methods is the lack of static (leakage) power consideration. It is well known that power of microprocessors is mainly composed of dynamic power and static (leakage) power. Dynamic power is caused by the logic gate switching, and can be estimated by obtaining the logic activity rate of each module using performance counter. Most existing runtime full-chip thermal estimation methods use dynamic power only without considering leakage power or just use simplified leakage power models. One reason is that dynamic power accounts for the majority of the total power for old IC technologies, thus considering dynamic power only is sufficient for runtime thermal estimation. Another reason is there exists the nonlinear relationship between leakage power and on-chip temperature. As a result, thermal estimation considering leakage power becomes nonlinear transient simulation process, which is difficult to compute and cannot scale to very large problem sizes (such as full-chip thermal analysis) for runtime applications.

However, for today's microprocessors, the leakage power cannot be neglected anymore in runtime thermal estimation as the percentage of leakage power in total power is quite

- *H. Wang, J. Wan, C. Zhang, and H. Tang are with State Key Laboratory of Electronic Thin Films and Integrated Devices, University of Electronic Science and Technology of China, Chengdu 610054, China, and with School of Microelectronics and Solid-State Electronics, University of Electronic Science and Technology of China, Chengdu 610054, China. E-mail: {wanghai, jiachunwan, zhangc, tanghe}@uestc.edu.cn.*
- *Sheldon X.-D. Tan is with Department of Electrical Engineering, University of California, Riverside, CA 92521. E-mail: stan@ee.ucr.edu.*
- *Y. Yuan is with School of Automation Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China. E-mail: yuany@uestc.edu.cn.*
- *K. Huang and Z. Zhang are with Southwest China Research Institute of Electronic Equipment, Chengdu 610036, China. E-mail: {kehenghuang, zhenghongzhang}@swiee.com.*

Manuscript received 15 Feb. 2017; revised 11 Nov. 2017; accepted 18 Nov. 2017. Date of publication 30 Nov. 2017; date of current version 13 Apr. 2018. (Corresponding author: Hai Wang.)
Recommended for acceptance by D. Atienza.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2017.2778066

significant for new generations of the microprocessors. The breakdown of Dennard scaling is a clear indication of this trend as leakage power does not scale with transistor size [1]. What is even worse is that leakage is exponentially dependent on temperature, so the leakage power will cause the processors to heat up and further increase leakage power itself. As a result, leakage power is one of the most important limiting factors of processor performance today and an important or even major part of total power for thermal estimation.

In order to take the important leakage power into account for runtime thermal estimation, we have to fully consider nonlinear interactions between leakage power and temperature. Existing iterative methods were proposed to handle such nonlinearity [9], [10] for steady state thermal estimation. Although these methods are considered to be accurate, they perform temperature calculation using thermal model multiple times in the iteration, which is slow for full-chip thermal estimation. The Green's function based technique was also proposed to handle the leakage-aware thermal estimation problem in [11]. However, it is unable to handle general transient thermal estimation. More discussions of relevant work will be given in the next section.

In this article, we propose a fast leakage-aware full-chip transient thermal estimation method. The new method tries to mitigate the mentioned problems in the existing full-chip thermal estimation methods. Our major contributions are summarized as follows:

(1) First, to avoid the iteration between thermal analysis and leakage analysis, the new method uses Taylor expansion based local linearization technique to build a number of localized linear thermal models. The new linear thermal models are formulated in traditional thermal model form, so that general simulation methods can be easily applied to them.

(2) To further increase the thermal estimation speed, a specially designed model order reduction method with partial and incremental SVD update technique is integrated into the estimation framework to generate local compact thermal models.

(3) Numerical results demonstrate that the new thermal analysis method is able to accurately estimate full-chip transient temperature distribution by fully considering the nonlinear leakage-temperature dependency and it is significantly faster than the iteration based method.

The remaining part of this article is organized as follows. In Section 2, we first review the important works in fast thermal estimation of IC systems. Next, in Section 3, we present the basic knowledges of static power modeling and thermal modeling techniques, and introduce the iteration based leakage-aware thermal estimation and its problems. Then, we demonstrate our new fast leakage-aware full-chip transient thermal estimation method in Section 4. The experimental results showing the accuracy and speed of the newly proposed method are presented in Section 5. Finally, Section 6 concludes this article.

2 Prior Work

In this section, we briefly review some important researches in fast thermal estimation of IC chips, especially in leakage-aware thermal estimation.

Many thermal estimation methods have been proposed to aid thermal aware design and runtime thermal regulation of the microprocessors. The thermal estimation using numerical finite element methods (FEM) or finite difference methods (FDM) such as ANSYS and COMSOL are quite accurate but very computationally expensive. These tools do not fit well for architectural level thermal aware design and runtime thermal regulation in which estimation speed and efficiency are critical. As a result, many efforts were devoted to accelerating the FEM/FDM based thermal estimation: ISAC utilizes the spatially adaptive thermal modeling technique to increase thermal estimation speed [7]; model order reduction based methods speed up thermal estimation by reducing the size of the original thermal model [12], [13], [14]; HotSpot simplifies the package and chip model by using compact RC based model [10], [15]. TILTS was developed based on HotSpot by assuming power remains constant between two adjacent discrete time points [16].

Besides FEM/FDM based methods, Green's function based methods were also proposed for full-chip thermal estimation using 2-D spatial Fourier transforms, such as the work in [17]. Unlike FEM/FDM based methods which usually have no problem at performing transient thermal simulation, Green's function based methods are mainly used for steady state thermal analysis [17]. To mitigate this problem, the Power Blurring method was developed based on the Green's function with transient thermal estimation ability [18].

The works mentioned above share a common problem: they have difficulty in considering the temperature dependent leakage power for transient thermal estimation, because they are based on linear thermal systems. Only few fast thermal estimation works are able to handle leakage power. For instance, HotSpot applies the iterative based method for steady state analysis with degraded simulation efficiency. In [19], researchers proposed a method to estimate leakage power using coarse-grained thermal models. However, this method does not have transient estimation ability, and the leakage power it provides is too coarse to be used for full-chip thermal estimation. Recently, LightSim [11] and 3DSim [20] try to provide a leakage-aware transient thermal estimation method based on Green's function. However, they are limited to calculating the step temperature response with only constant (time invariant) input power map, and is not able to perform general transient thermal estimation with time varying power map traces.

3 Background

In this section, we first present static power modeling and thermal modeling techniques, which are important basic knowledges for our new work. Then, we show the traditional iteration based solution of the leakage-aware thermal estimation, and point out its problems, which are solved in this work. The mathematical notations used in this article are summarized in Table 1 for better presentation.

3.1 Static Power Modeling

It is well known that, the total power of chip, denoted as p, is composed of dynamic power and static power. The dynamic power, denoted as p_d, depends on the activity of the chip, and thus can be easily estimated by performance counter based methods [21], [22], [23].

TABLE 1
Mathematical Notations

p, P	total power in scalar form and vector form
p_d, P_d	dynamic power in scalar form and vector form
p_s, P_s	static power in scalar form and vector form
I_{leak}	total leakage current
I_{sub}, I_{gate}	subthreshold current and gate leakage current
I_{lin}	linearized subthreshold current
v_T	thermal voltage
T_p, T	temperature in scalar form and vector form
T_{p_0}	Taylor expansion temperature point
K, η	process related parameters for leakage current
P_0, A_s	vector and matrix for linear static power model (9)
G, C, B, L	thermal model matrices of the whole system
Y	temperature vector with only chip temperatures
G_l	new G matrix for linearized thermal model
M	sampling response matrix used for MOR
M_a	new M at new Taylor expansion points
M_L	sampling response matrix with both M and M_a
U, Σ, V	SVD matrices of M as in (15)
U_t, Σ_t, V_t	SVD matrices inside incremental SVD
F, H, U_L, Σ_L	temporary matrices inside incremental SVD
Q, R	QR factorization matrices inside incremental SVD
U_r	the projection matrix in MOR
$\hat{G}_l$, $\hat{C}$, $\hat{B}$, $\hat{L}$	reduced linearized thermal model matrices
$\hat{T}$	temperature vector in the reduced thermal model

Very different from dynamic power, the static power p_s, caused by leakage current I_{leak} as

$$p_s = V_{dd} I_{leak}, \tag{1}$$

is independent of the chip's activity. Values of static power are harder to obtain than dynamic power, mainly because of the special temperature sensitivity caused by leakage current. IC leakage current has various components, including subthreshold current, gate current, reverse-biased junction leakage current and so on. Among these components, subthreshold current I_{sub} and gate leakage current I_{gate} are the main parts. As a result, we can ignore other parts of leakage and get the leakage current approximation [19], [24], [25] as

$$I_{leak} = I_{sub} + I_{gate}. \tag{2}$$

The subthreshold current is modeled in the commonly accepted MOSFET transistor model BSIM 4 [26] as (also apply $V_{DS} \gg v_T$ [19])

$$I_{sub} = K {v_T}^2 e^{\frac{V_{GS}-V_{th}}{\eta v_T}} \left(1 - e^{\frac{-V_{DS}}{v_T}}\right) \approx K {v_T}^2 e^{\frac{V_{GS}-V_{th}}{\eta v_T}}, \tag{3}$$

where $v_T = \frac{kT_p}{q}$ is the thermal voltage and T_p is a scalar representing temperature at one place,[1] K and η are process related parameters, and V_{th} is the threshold voltage.

While the subthreshold current is highly related to temperature, the gate current I_{gate}, which results from tunneling between the gate terminal and the other three terminals, does not depend on temperature and can be considered as a technology-dependent constant.

Apparently, the leakage current has a complex relationship with temperature. In this work, we use (1), (2), and (3) to model the static power considering such relationship. The parameters of leakage current can be obtained by curve fitting using HSPICE simulation data. In order to see the accuracy of the model used, Fig. 1 shows an HSPICE simulation result of leakage using TSMC 65nm process model and its curve fitting result using approximate leakage model. From the figure, we can see that the static power model has high accuracy for all common temperatures of IC chips.

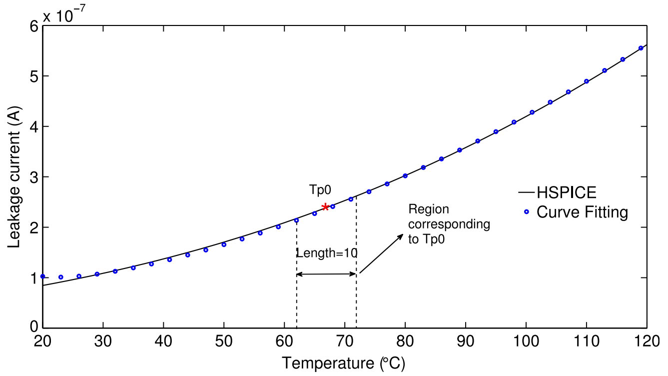


Fig. 1. Comparison of leakage of a TSMC 65 nm process MOSFET from HSPICE simulation with its curve fitting result using (3). An example of temperature region division is also shown in the figure, which will be discussed later.

We can conclude that the static power distribution depends mainly on the temperature distribution for a certain chip with constant physical parameters. Since temperature also depends on power, in order to view the whole picture, thermal model of IC chip is used to describe temperature's dependency on power as shown next.

3.2 Thermal Modeling

In order to calculate the full-chip temperature distribution, a thermal model with the ability to link the power and temperature is needed. To perform thermal analysis for an IC chip, we usually divide both the chip and its package into multiple blocks called thermal nodes, with the partition granularity determined by accuracy requirements. Then we compute the thermal resistances and capacitances among these thermal nodes, which model the thermal transport and power response behaviors.

For example, for a certain chip with n total thermal nodes, we can generate its thermal model as

$$\begin{aligned} GT(t) + C\frac{dT(t)}{dt} &= BP(T,t), \\ Y(t) &= LT(t), \end{aligned} \tag{4}$$

where $T(t) \in \mathbb{R}^n$ is the temperature vector (distinguished from T_p, which is a scalar representing temperature at only one place), representing temperatures at n places of the chip and package; $G \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{n\times n}$ contain equivalent thermal resistance and capacitance information respectively; $B \in \mathbb{R}^{n\times l}$ stores the information of how powers are injected into the thermal nodes; $P(T,t) \in \mathbb{R}^l$ is the power vector, which contains power consumptions of l components on chip, including both dynamic power vector P_d and static power vector P_s, i.e., $P(T,t) = P_s(T,t) + P_d(t)$, reminding that static power $P_s(T,t)$ is actually a function of temperature T; $Y(t) \in \mathbb{R}^m$ is the output temperature vector, containing only temperatures of thermal nodes that the user

1. T introduced latter in (4) is a vector representing temperatures at multiple positions.

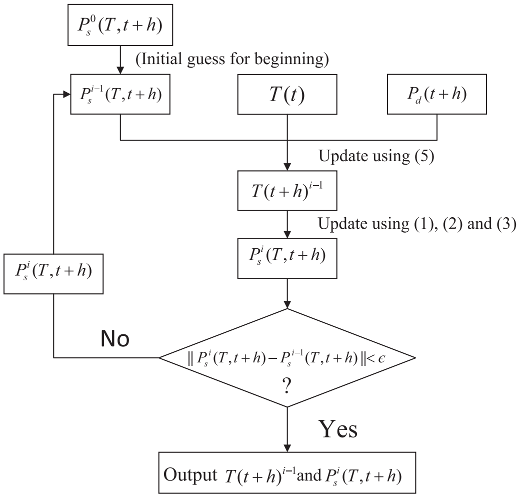


Fig. 2. Flow diagram of traditional iteration based transient thermal estimation method for one time step.

is interested in, for example, thermal nodes on the chip only (excluding package thermal nodes); $L \in \mathbb{R}^{m\times n}$ is the corresponding output selection matrix which selects the m chip temperatures from $T(t)$.

Model (4) successfully links power and temperature distribution of chip, but for computer based simulation, we still have to take care of the differential term "$dT(t)/dt$". Normally, we can discretize it, for example using backward Euler's method with time step h as

$$\left(\frac{C}{h}+G\right)T(t+h)=\frac{C}{h}T(t)+B(P_d(t+h)+P_s(T,t+h)). \quad (5)$$

The next question would be how we can use the thermal model (5) for transient thermal estimation. It seems straightforward as in the flowing way: simply take $T(t)$ (previously calculated, or initial value provided) together with the power information, and we are able to calculate the state of the next time step $T(t+h)$. However, such transient thermal estimation is valid for dynamic power only scenario and cannot be used if static power is considered. This is because the static power $P_s(T,t+h)$ is a function of current temperature $T(t+h)$, leading to the fact that we need $T(t+h)$ to compute $P_s(T,t+h)$ while we also need $P_s(T,t+h)$ to compute $T(t+h)$, similar to the famous chicken or the egg causality dilemma.

3.3 Iteration Based Leakage-Aware Thermal Estimation

As explained before, due to the dependency of static power on temperature, (5) is a nonlinear equation, and as a result, $T(t+h)$ cannot be calculated directly. Traditionally, iterative method is used to solve such equation [9], [20], [27], using the flow for one thermal estimation time step shown in Fig. 2.

First, based on the process technology used, we determine $P_s^0(T,t+h)$, which is an initial guess of $P_s(T,t+h)$. Temperature distribution $T(t+h)^0$ is calculated using (5) with such initial guess. Then, the static power is updated as $P_s^1(T,t+h)$ using (1), (2), (3) with $T(t+h)^0$. Next, temperature distribution can be updated again as $T(t+h)^1$ using thermal model (5) and $P_s^1(T,t+h)$, which concludes one iteration loop. Such iteration goes on until the convergence test is satisfied as $\|P_s^i(T,t+h)-P_s^{i-1}(T,t+h)\|<\epsilon$. Finally, $T(t+h)^{i-1}$ is outputted as the estimation result for the current time step.

Although the result of this iteration based method is considered to be accurate when the tolerance ϵ is chosen to be small enough, its computing time is a serious problem. For full-chip leakage-aware temperature estimation, thermal model in (5) is large, especially when a fine-grained chip thermal analysis is needed. Solving (5) many times at each time step makes the simulation time to be long, which is a drawback of the iteration based method when it is used for runtime temperature estimation.

4 Fast Leakage-Aware Full-Chip Thermal Estimation

In this work, in order to resolve the long computing time problem of the iteration based method, we propose a novel non-iteration based fast full-chip temperature estimation method, which adaptively transforms the original nonlinear thermal model into local linear thermal model to avoid the time-consuming iterations. In addition, an adaptive model order reduction method with incremental SVD update technique is specially designed and integrated into the non-iteration based thermal estimation method to achieve further speedup.

4.1 Local Linearization of Subthreshold Current

As shown before, the major difficulty of calculating leakage-aware temperature distribution comes from the nonlinear thermal model shown in (5), which is caused by the nonlinear dependency of subthreshold current on temperature. Thus, a basic idea of our proposed method is to approximate the original nonlinear leakage model using multiple new linear leakage models. By using such linear leakage models, we can reformulate the original nonlinear thermal model into multiple linear thermal models, such that traditional non-iteration based thermal estimation method can be applied to them.

In order to generate a linear leakage model, we perform Taylor expansion on the original nonlinear I_{sub} model (3) at a reference temperature point T_{p0} as

$$\begin{aligned} I_{sub} =&K\left(\frac{k}{q}\right)^2 e^{\frac{q(V_{GS}-V_{th})}{\eta k T_{p0}}} \\ &\times\left({T_{p0}}^2+\left(2T_{p0}-\frac{q(V_{GS}-V_{th})}{\eta k}\right)(T_p-T_{p0})\right) \\ &+o[(T_p-T_{p0})^2], \end{aligned} \quad (6)$$

where $o[(T_p-T_{p0})^2]$ is the remainder. If we approximate the original function by ignoring the remainder $o[(T_p-T_{p0})^2]$, we can then get the linearized I_{sub}, denoted as I_{lin}

$$\begin{aligned} I_{lin} =&K\left(\frac{k}{q}\right)^2 e^{\frac{q(V_{GS}-V_{th})}{\eta k T_{p0}}} \\ &\times\left({T_{p0}}^2+\left(2T_{p0}-\frac{q(V_{GS}-V_{th})}{\eta k}\right)(T_p-T_{p0})\right). \end{aligned} \quad (7)$$

Normally, the approximation accuracy of I_{lin} can be guaranteed when the reference temperature point T_{p0} is close enough to the actual temperature value T_p. From previous research, it has been shown that due to the characteristics of today's semiconductor process, such local linear approximation of leakage has high accuracy around the expansion temperature point [11], [19].

4.2 Formulating Local Linear Thermal Model

Since we have linearized the relation of subthreshold current and temperature, we can rewrite the static power and temperature relation in a linear form as

$$\begin{aligned} p_s &= V_{dd} I_{leak} \\ &= V_{dd} \times (I_{lin} + I_{gate}) \\ &= V_{dd} \times (I_{lin}(T_p) + I_{const}), \end{aligned} \tag{8}$$

where $I_{lin}(T_p)$ represents the terms associated with T_p in (7), I_{const} contains constant terms that are not associated with T_p in (7) and the gate leakage I_{gate}.

Based on this new static power model, we can rebuild a linear thermal model to replace (5). In order to do that, we need to integrate (8) into (4). Please note that (8) is in scalar form for only one certain thermal node while (4) is in vector/matrix form including information of all thermal nodes. So we first rewrite (8) in vector/matrix form by collecting and accumulating scalars $I_{lin}(T_p)$ and I_{const} at multiple positions of the chip into vectors, then change the current variables to power by multiplying voltage V_{dd}. Rewriting from (8), the linearized static power representation in vector/matrix form is

$$P_s = P_0 + A_s T, \tag{9}$$

where $P_0 \in \mathbb{R}^l$ is a known vector, with each element formed by terms not associated with T_p in (8) at each position of the chip. $A_s \in \mathbb{R}^{l \times n}$ is a known rectangular diagonal matrix (the left $l \times l$ block matrix is diagonal representing thermal nodes on the chip, and the right $l \times (n-l)$ block matrix is all zeros representing the thermal nodes of package), with each diagonal element formed by the coefficient associated with T_p in (8) at each position of the chip.

Integrating (9) into (4), and let $G_l = G - BA_s$, we have

$$\begin{aligned} G_l T(t) + C\frac{dT(t)}{dt} &= B(P_d(t) + P_0), \\ Y(t) &= LT(t). \end{aligned} \tag{10}$$

Now, we have successfully obtained a linear thermal model considering static power and eliminated the nonlinear relationship of static power and temperature. Then, we can discrete this model using backward Euler's method, resulting in its transient estimation form similar to (5) as

$$\begin{aligned} \left(\frac{C}{h} + G_l\right) T(t+h) &= \frac{C}{h} T(t) + B(P_d(t+h) + P_0), \\ Y(t+h) &= LT(t+h). \end{aligned} \tag{11}$$

Obviously, simulating the locally linearized leakage-aware thermal model is as straightforward as in (5) by viewing "G_l" as the new "G" matrix, and "$P_d(t) + P_0$" as the new "$P(T,t)$" vector.

4.3 Selecting the Proper Expansion Points

Although the new linear thermal model can be generated as shown before, the Taylor expansion temperature points still need to be determined since the linear thermal model accuracy depends on them, and P_0 and A_s in (9) are formulated by the expansion point information. Now, we discuss how to choose proper values of T_{p_0} for thermal nodes on the chip.

As shown in Section 4.1, as a property of Taylor expansion approximation, linear approximation (7) (also the equivalent (9) and (10)) is accurate if the actual temperature T_p (or T in vector form) is close enough to T_{p_0}. As a result, in order to ensure the approximation accuracy, we want each expansion point T_{p_0} to be close to the actual temperature T_p in transient thermal estimation. This means that the straightforward choice of an expansion point is $T_{p_0} = T_p$. However, such strategy requires updating T_{p_0} at each time step, leading to long computing time because many LU decompositions have to be performed. To see this problem clearly, please note that we need to perform LU decomposition of $(\frac{C}{h} + G - BA_s)$ in the transient thermal estimation process in (11), and matrix A_s depends on the Taylor expansion points T_{p_0}. If we update the expansion points for every estimation time step, LU decomposition also has to be re-performed for every time step, causing serious computing cost problem.

In order to balance the accuracy and computing cost, we need to propose a flexible strategy to update the Taylor expansion point T_{p_0}. By observing Fig. 1, we notice that at positions where the nonlinearity of I_{sub} is relatively weak, $o[(T_p - T_{p0})^2]$ can be small even if T_{p_0} is far from T_p. Inspired by this, we propose a strategy to determine Taylor expansion points in transient analysis: for each temperature, we set a temperature region with a certain length, as shown in Fig. 1. Assume T_{p_0} is taken as the Taylor expansion point for a thermal node, such expansion point T_{p_0} will be used when the node temperature T_p is within the temperature region of T_{p_0} (in Fig. 1, it is the region with 10°C length as example). We may update the expansion point only when the node temperature T_p is out of the temperature region of T_{p_0}. The temperature region lengths are determined off-line according to the nonlinear temperature-leakage curve of a specific fabrication process to balance the estimation accuracy and speed. In general, shorter temperature region leads to better accuracy but slower speed for estimation, as shown later in experiments (Section 5.4). In addition, the region can be shorter for temperature point with stronger nonlinearity, and vice versa. For the temperature-leakage curve shown in Fig. 1, the strengths of the nonlinearity are quite similar for the whole temperature range, so we simply use the same region length for all temperatures.

It is also noticed that T_p is an unknown variable. Thus, we need some available information to replace T_p, in order to determine the correct temperature regions and the corresponding Taylor expansion points. In this work, we employ the on-chip physical thermal sensors to achieve such purpose. Since there are only limited number of thermal sensors and we also do not want to change the linearized model (10) (A_s and P_0) for temperature region change at single or very few positions, we use the thermal sensor readings to test our estimation error in real-time and determine whether we should change the linearized model or not. Assume there are k thermal sensors with readings at current time as $T_{sen_1}, T_{sen_2}, \ldots, T_{sen_k}$, and the corresponding estimated temperature values by (10) at thermal sensor positions are $T_{est_1}, T_{est_2}, \ldots, T_{est_k}$. Then the maximum estimation error at the sensor positions is calculated as

$$Err_{max} = \max_{i=1,2,\ldots,k} |T_{sen_i} - T_{est_i}|. \tag{12}$$

If $Err_{max} > Err_{th}$, where Err_{th} is the user defined threshold value, it means that the current linear thermal model (10) is not accurate any more, as demonstrated experimentally in Fig. 6. In this case, we update the linear thermal model by changing the Taylor expansion points for all thermal nodes based on their temperature regions as shown in Fig. 1. Otherwise, we just keep using the current linear thermal model and continue the transient thermal estimation process.

Besides being able to estimate temperature at runtime with thermal sensors, the new method can also be used at design time when thermal sensor is unavailable. As a result, we propose a thermal estimation solution even without thermal sensor information: we simply estimate the temperature regions using the temperature distribution at previous time point and determine expansion points for all nodes. Testing this strategy with "lin & svd update" on the 16-core system as shown in the experiment section, we still get a good result in both accuracy and speed: an average temperature estimation error of $0.32^\circ\mathrm{C}$ and an average speedup of $56.64\times$ against "ite" and $11.39\times$ against TILTS.

4.4 Speed up Thermal Estimation by Model Order Reduction with Incremental SVD Update

Although we have successfully obtained linear thermal models to avoid iterations, the size of the linear thermal model in (10) is large especially when fine-grained thermal analysis is performed. One may naturally assume that traditional model order reduction (MOR) can be applied to the linear thermal model (10) to further speed up thermal estimation. Unfortunately, such simple strategy does not work well for the leakage-aware thermal estimation. The reason is that the linear thermal model keeps changing during thermal estimation process due to the change in Taylor expansion points introduced previously. In order to handle that, one may offer two solution choices, but neither of them will work. One choice is to re-perform MOR upon the change of linear thermal model. However, by taking this solution, we may end up with limited speedup or even longer estimation time because MOR process itself takes a lot of time as it requires solving the original linear thermal model. Another choice is to perform MOR offline for all possible linear thermal models, and store the reduced models in a library for online thermal estimation usage. However, since each position on chip has multiple potential temperature regions, the number of possible linear thermal models is extremely large. Performing MOR on all these possible linear thermal models offline and storing them in a library is impossible for both computing time and storage aspects.

In this section, we propose a model order reduction method specially designed for our new leakage-aware thermal estimation method. It updates the projection matrix of MOR for only few necessary scenarios. The projection matrix update is also performed in an incremental way, which greatly reduces the update number and MOR computing time. We will first introduce how MOR is applied to single linear thermal model, then the proposed MOR method for the leakage-aware thermal estimation is presented.

4.4.1 Model Order Reduction for Single Linear Thermal Model

First, we show how MOR can be performed on single linear thermal model (10) to generate a compact thermal model. Modern MOR methods are mostly projection based [28], [29]. The basic idea of these projection based methods is to pass the important information of the original model to the reduced model through the projection process. Depending on which information is passed to the reduced model, projection based MOR methods can be classified into several categories. In this article, we demonstrate the popular sampling based MOR method, which passes original model's state frequency responses of several frequency points (called sampling points) to the reduced model through projection. Please note that many other MOR methods can also be used.

To formulate the projection matrix, we need to compute the state frequency responses of the thermal model (10) at the sampling points. Assume we choose several sampling points $s_1, s_2, \ldots, s_p$, for the ith sampling point, we calculate the corresponding state frequency response of (10) as

$$T(s_i) = (G_l + s_i C)^{-1} B, \tag{13}$$

where $T(s_i) \in \mathbb{R}^{n\times l}$.[2] By collecting frequency responses of all sampling points, we can generate a sampling response matrix as

$$M = [\,T(s_1)\ T(s_2)\ \ldots\ T(s_p)\,], \tag{14}$$

where $M \in \mathbb{R}^{n\times lp}$.

On the sampling point selection side, we suggest choose more low frequency sampling points for thermal models. This is because the equivalent thermal circuit works as a low pass filter, most high frequency component on the power side will be filtered out on the temperature side. In another word, temperature does not have much high frequency component even when the input power is changing fast. We have plotted the frequency responses of the original thermal model and the reduced model generated using low frequency samples in experiment part (Fig. 8). It is clear that the frequency responses show low pass filter properties, and the reduced model only shows noticeable error beyond 100 Hz, which already has a huge magnitude drop from DC. We also remark that sampling points can be found in an adaptive and automatic way to optimize the wide frequency band accuracy. For details of such automatic sampling point selection methods, please refer to [30].

It is noted that M may contain a lot of redundant information, since frequency responses of two different sampling points may contain similar information. In order to get rid of the redundancies, we further perform singular value decomposition (SVD) on M as

$$U\Sigma V^T \overset{\text{SVD}}{\longleftarrow} M \tag{15}$$

where $U \in \mathbb{R}^{n\times n}$ is a unitary matrix whose columns span the *column space* of M, $V \in \mathbb{R}^{lp\times lp}$ is also a unitary matrix whose columns span the *row space* of M, $\Sigma \in \mathbb{R}^{n\times lp}$ is a diagonal matrix with non-negative singular values σ_i listed in

2. Since we want the reduced model to be compatible with all different input combinations, we replace the input vector with an identity matrix.

descending order on the diagonal. One property of SVD is that the singular value reveals the importance of its corresponding column space basis in U as well as its row space basis in V. In another word, if σ_i is very small, we can simply eliminate σ_i as well as the ith columns of U and V in (15), and still get a quite good approximation of M, meaning information in ith columns of U and V is redundant.

The fact that singular values are ordered in Σ means columns of U are ordered by importance. So, we only need to keep the first q columns of U and eliminate other columns without losing much accuracy, as

$$U_r = [\, u_1 \; u_2 \; \dots \; u_q \,], \tag{16}$$

where u_i is the ith column of U. U_r retains the most important column space information of M. So, we can use U_r as the projection matrix, and generate a reduced model as

$$\begin{aligned} \hat{G}_l \hat{T}(t) + \hat{C}\frac{d\hat{T}(t)}{dt} &= \hat{B}(P_d(t) + P_0), \\ Y(t) &= \hat{L}\hat{T}(t), \end{aligned} \tag{17}$$

where $\hat{G}_l \in \mathbb{R}^{q\times q}$, $\hat{C} \in \mathbb{R}^{q\times q}$, $\hat{B} \in \mathbb{R}^{q\times l}$, and $\hat{L} \in \mathbb{R}^{m\times q}$ are calculated as

$$\hat{G}_l = U_r^T G_l U_r, \hat{C} = U_r^T C U_r, \hat{B} = U_r^T B, \hat{L} = L U_r. \tag{18}$$

4.4.2 Thermal Estimation Using Reduced Local Linear Thermal Models

Performing thermal estimation using single reduced thermal model is easy with the following transient simulation form

$$\begin{aligned} \left(\frac{\hat{C}}{h} + \hat{G}_l\right)\hat{T}(t+h) &= \frac{\hat{C}}{h}\hat{T}(t) + \hat{B}(P_d(t+h) + P_0), \\ Y(t+h) &= \hat{L}\hat{T}(t+h). \end{aligned} \tag{19}$$

However, in our situation, we obtain a series of reduced local linear thermal models of different Taylor expansion points on the fly in thermal estimation procedure. Since these reduced local linear thermal models may be generated by different projection matrix U_r, the direct thermal estimation method shown above is invalid at the time when the projection matrix U_r is updated.

To see this problem clearly, assume we update the projection matrix U_r at current time $t+h$. Let the projection matrix of previous time step t as $U_r^{(p)}$ and the calculated reduced temperature state at previous time step as $\hat{T}^{(p)}(t)$. Now if we calculate the reduced temperature state at current time $\hat{T}^{(c)}(t+h)$ by directly using (19), we get

$$\begin{aligned} &\left(\frac{\hat{C}^{(c)}}{h} + \hat{G}_l^{(c)}\right)\hat{T}^{(c)}(t+h) \\ &= \frac{\hat{C}^{(c)}}{h}\hat{T}^{(p)}(t) + \hat{B}^{(c)}(P_d(t+h) + P_0), \end{aligned} \tag{20}$$

where $\hat{G}_l^{(c)}$, $\hat{C}^{(c)}$, and $\hat{B}^{(c)}$ are reduced models calculated using current projection matrix $U_r^{(c)}$. Obviously, Equation (20) shown above is incorrect, because current reduced model is generated by the new projection matrix $U_r^{(c)}$, while $\hat{T}^{(p)}(t)$ is calculated by previous reduced model which is generated by $U_r^{(p)}$. Please note that reduced temperature states $\hat{T}^{(p)}(t)$ and $\hat{T}^{(c)}(t)$ can be totally different, because MOR only retains accuracy of the output ($Y(t)$ in (17)), but not the state ($\hat{T}(t)$ in (17)).

To solve such problem, we need to transform the previously calculated $\hat{T}^{(p)}(t)$ into the new subspace spanned by the new projection matrix $U_r^{(c)}$, so we can get the approximation of $\hat{T}^{(c)}(t)$. Because we have the following approximation

$$T(t) \approx U_r \hat{T}(t), \tag{21}$$

which is applicable to both $\hat{T}^{(c)}(t)$ and $\hat{T}^{(p)}(t)$ with $U_r^{(c)}$ and $U_r^{(p)}$, respectively, we can transform $\hat{T}^{(p)}(t)$ into current subspace to approximate $\hat{T}^{(c)}(t)$ as

$$\hat{T}^{(c)}(t) \approx (U_r^{(c)})^{\dagger} U_r^{(p)} \hat{T}^{(p)}(t), \tag{22}$$

where the symbol "†" means pseudo inverse.

Now, we modify (20) into the following form as

$$\begin{aligned} \left(\frac{\hat{C}^{(c)}}{h} + \hat{G}_l^{(c)}\right)\hat{T}^{(c)}(t+h) &= \frac{\hat{C}^{(c)}}{h}(U_r^{(c)})^{\dagger} U_r^{(p)} \hat{T}^{(p)}(t) \\ &\quad + \hat{B}^{(c)}(P_d(t+h) + P_0). \end{aligned} \tag{23}$$

This equation is used to handle the projection matrix update in thermal estimation using reduced local linear models.

4.4.3 Specially Designed Model Order Reduction with Incremental SVD Update

Now, we are able to generate compact thermal models using previously presented method for transient thermal analysis. However, because G_l depends on the Taylor expansion points and U_r is generated using G_l, the whole MOR process introduced previously needs to be re-performed if the original linear thermal model is updated due to the change in expansion points. Obviously, we do not want to re-perform the full MOR process every time the original model is updated. Instead, we propose two methods to greatly reduce the thermal estimation overhead caused by MOR.

The first method prevents thermal estimation from unnecessary updating of the projection matrix U_r in MOR when the linear thermal model is updated. It is based on the observation that one projection matrix may work for several thermal models provided that the differences among these thermal models are limited. To explain more clearly, assume the projection matrix U_r is generated by sampling a thermal model "Model A". Then columns of U_r span a subspace which includes all sampling points' frequency responses of "Model A". For another thermal model "Model B" with limited difference from "Model A", the responses of important frequencies of "Model B" may be still inside or not far away from the subspace spanned by "Model A". Thus, we can still use U_r generated by sampling "Model A" to reduce "Model B".

Based on the discussion above, during the thermal estimation process, even when the thermal model is updated due to the expansion points change, we may keep on using the previous projection matrix for the updated thermal model if that projection matrix is determined to be still accurate. The detailed operation is presented as follows. When thermal model (19) needs to be updated due to expansion points change, we first update matrix $G_l = G - BA_s$, and then generate its corresponding reduced matrix $\hat{G}_l$ using

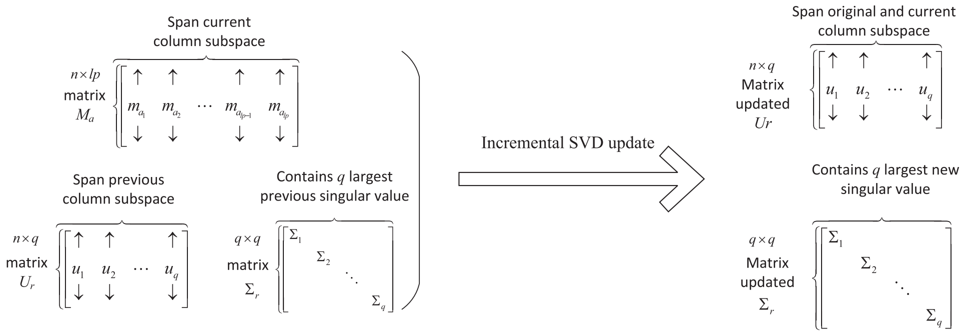


Fig. 3. The diagram of updating U_r using the incremental SVD update method.

the *previous* projection matrix U_r. Next, we calculate Err_{max} using the temperature estimation results from the reduced model with new $\hat{G}_l$. If Err_{max} is smaller than Err_{th}, it means the projection matrix U_r is still valid and we continue the thermal estimation using the updated reduced model. Computing $\hat{G}_l$ with previous U_r requires only one matrix multiplication, so the cost is very low. Instead, if Err_{max} is larger than Err_{th}, meaning U_r does not work well for the updated thermal model. In this situation, we need to compute a new projection matrix U_r using the updated thermal model.

The second method takes action when the first method determines that the previous projection matrix U_r is not valid anymore and should be updated. In that case, the second method seeks for partial updating technique for the projection matrix U_r instead of re-computing U_r from draft. Such strategy comes from two reasons. First, if we compute U_r from draft, we need to re-perform the whole MOR process including solving the new original model at the sampling points and totally re-doing SVD of the new sampling response matrix M. This will result in large computing cost. Second, although the projection matrix U_r is not good enough for current temperature ranges (as determined in the first method), it still contains important information for some other temperature ranges (especially around the former temperature ranges, from which it is computed). Because it is common that similar temperature distributions may appear many times during the thermal estimation process, throwing away all information of the previous U_r and generating a completely new projection matrix considering only current situation is not an ideal choice. To further increase the efficiency of MOR, we propose a method to update U_r partially and incrementally as shown next.

To keep the past useful information at updating the projection matrix, we first calculate a new sampling response matrix, denoted as M_a, using the updated linear thermal model generated from the new Taylor expansion points. Then we append M_a to the previous response matrix M and get a larger matrix as $M_L = [\,M \quad M_a\,]$. M_L contains both current and previous thermal model information, i.e., M_L is able to reduce both current and previous thermal distributions (Taylor expansion points) induced thermal models with good accuracy.

Obviously, there are redundant information between M and M_a. In order to remove such redundancy and generate a new compact projection matrix that covers important information of M_L, one natural idea is to directly perform SVD similar to previously introduced steps (15) and (16). However, this leads to high computing cost.

A partial and incremental SVD update method can be used instead of the original SVD process to deal with these problems with much faster speed [31]. At the beginning of thermal estimation process, we compute the first U_r using (15) and (16). Additionally, we also keep the truncated singular value matrix Σ_r with only first q singular values. During thermal estimation process, when U_r becomes invalid, we use these two matrices U_r and Σ_r, together with the new sampling response matrix M_a, to compute a new diagonal matrix Σ_r and left singular matrix U_r as shown in Fig. 3 and presented in the following. Let $F = U_r^T M_a$, $H = M_a - U_r F$, and $QR \overset{\text{QR}}{\leftarrow} H$, where $\overset{\text{QR}}{\leftarrow}$ denotes QR factorization. We simply do SVD on a new boarded diagonal matrix as

$$U_t \Sigma_t V_t \overset{\text{SVD}}{\longleftarrow} \begin{bmatrix} \Sigma_r & F \\ 0 & R \end{bmatrix}, \tag{24}$$

and the new U_L and Σ_L are computed as

$$U_L = [\,U_r\ Q\,]U_t, \quad \Sigma_L = \Sigma_t. \tag{25}$$

Columns of U_L are orthogonal column space basis of $[\,U_r \quad M_a\,]$, which contain both previous and current sampling point response information. More importantly, columns of U_L are also sorted in importance, revealed by the diagonal values in Σ_L. As a result, we truncate U_L and Σ_L to the order of q, generating the new U_r and Σ_r matrices as

$$U_r \leftarrow U_L(:, 1:q), \quad \Sigma_r \leftarrow \Sigma_L(1:q, 1:q), \tag{26}$$

where we borrowed Matlab-like expression to denote the truncation process. Practically, in order to further reduce computing time, we can perform the truncation one step earlier at the SVD stage in (24). Obviously, the new U_r contains important information of both previous and current thermal models, with redundancies between them removed.

The flow of the transient leakage-aware thermal analysis with compact thermal model and specially designed projection matrix update strategy for one time step is summarized using flow chart shown in Fig. 4.

5 Experimental Results

In this section, we evaluate both accuracy and efficiency of the proposed fast full-chip leakage-aware transient thermal estimation technique.

5.1 Experiment Setup

First, we characterize the impact of temperature on device leakage through HSPICE simulation. Based on the

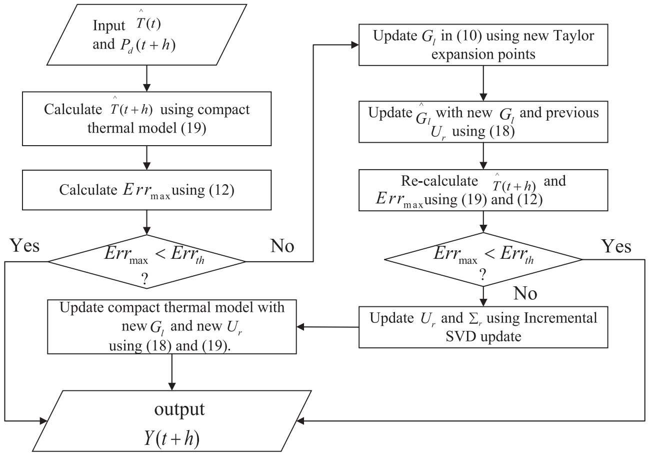


Fig. 4. Flow chart of our fast leakage-aware full-chip transient thermal estimation method for one time step.

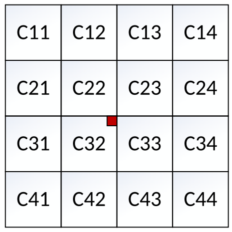


Fig. 5. Configuration of the 16-core chip. We put a probe grid (red square near the center) to demonstrate the transient temperature and power results.

simulation data, we obtain the parameters of model (3) through curve fitting as shown in Fig. 1. HotSpot 5.02 [15] is used to build the thermal models of four different chips with number of cores ranging from 9 to 36. The configuration of the 16-core chip is shown in Fig. 5 as an example. Sizes of all tested chips are 10 mm $\times$ 10 mm. Power estimator Wattch [32] is used to generate the dynamic power and instruction information by running the standard SPEC benchmarks. We use different power traces from SPEC benchmarks as the dynamic power traces of different cores on the chips. The ambient temperature is set to be $40^{\circ}\mathrm{C}$.

For thermal models of chips, we partition each core into 5×5 thermal blocks for fine-grained analysis, which results in $n = 912$ to 3612 total thermal nodes (including package thermal nodes) and $m = 225$ to 900 total on-chip thermal nodes (excluding package thermal nodes), for systems with core numbers ranging from 9 to 36. For MOR in this experiment, we reduce the original models with orders $n = 912$ to 3612 into reduced models with much smaller orders $q = 24$ to 50, with three sampling points $s_1 = 0$, $s_2 = \pm 0.001j$, $s_3 = \pm 0.1j$. The whole duration of the transient thermal estimation processes for all tests are 120s. In order to test the proposed methods under a wide range of temperatures, we scale the input dynamic power with different ratios during the 120s estimation period. For error control in our proposed methods, we set the threshold value Err_{th} to be $1^{\circ}\mathrm{C}$ and the length of temperature region to be $10^{\circ}\mathrm{C}$ for all tests unless specially noted.

For accuracy and speed comparison, we first perform the iteration based thermal estimation (which is accurate but time consuming as shown in Section 3.3) with *extra fine* estimation time step 0.001s and power trace sampling interval 0.001s to serve as the golden accuracy baseline (called "golden" in short). Then, two existing methods, the iteration based method (called "ite" in short) and TILTS [16], are used as the comparison counterparts of the new method. In order to be fair, all three methods ("ite", TILTS, and the new method) share the same time step 0.01s. They also share the same power trace which is averaged every 0.01s from the power trace used in "golden".

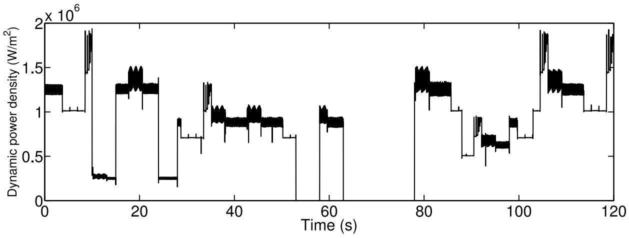
We compare the proposed method with TILTS because it speeds up HotSpot by assuming power remains constant between two adjacent discrete time points. However, TILTS is not able to deal with the nonlinear relationship between leakage power and temperature [16]. In addition, TILTS can be neither improved into an iteration based framework (like "ite" modified from HotSpot) nor an adaptive Taylor expansion based framework (like our new method) to handle leakage's nonlinear effect. In order to retain its original fast speed while still considering leakage/temperature dependency, we improve TILTS by linearizing leakage at a single Taylor expansion point using (9) to generate a new linear system like (10). Since TILTS can only be built based on one Taylor expansion point, we choose such point as $80^{\circ}\mathrm{C}$ which is the middle point of our test temperature range (from $40^{\circ}\mathrm{C}$ to $120^{\circ}\mathrm{C}$). Matrices of TILTS with 0.01s time step are computed offline using HotSpot simulation with 0.0001s time step, using procedure presented in [16].

Because our complete method includes three techniques (Taylor expansion based linear thermal models, applying MOR to the linear models on the fly, and using incremental SVD update to MOR), we use three cases, each with one more new technique than the previous one, to better analyze our proposed method. Specially, we use "lin only" to represent the first case, i.e., using Taylor expansion based linear thermal models only without MOR involved. "lin & svd batch" is used to represent further applying MOR to the linear thermal models and totally re-performing SVD to new sampling response matrix. "lin & svd update" represents the ultimate form of our newly proposed method, using all three techniques, i.e., using Taylor expansion based linear thermal models with incremental SVD update based MOR. As mentioned previously, we also use "golden" to represent the iteration based method with extra small simulation time step (0.0001s) and "ite" to represent the iteration based method with the normal simulation time step (0.01 s, shared by all methods in comparison) in figures and tables.

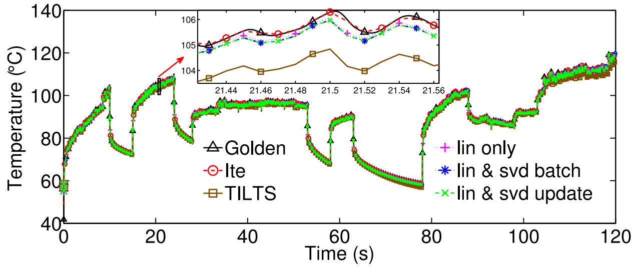
5.2 Estimation Accuracy of the Proposed Method

We first test the accuracy of the proposed thermal estimation method. Here we use the 16-core chip as an example for demonstration and discussion. Results on other chips are also collected and will be discussed later.

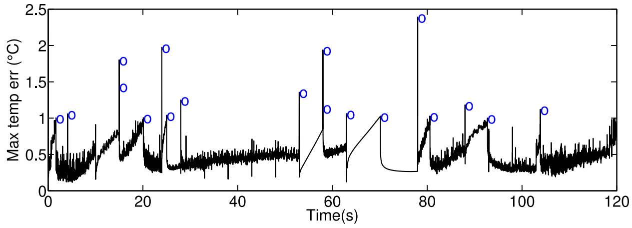
In order to demonstrate the transient thermal estimation results, we choose one grid near the center of the chip (marked as the red square in Fig. 5) to be the probe grid. The probe grid's input dynamic power during the transient thermal estimation process is shown in Fig. 6a. The estimated temperature results at the probe grid are shown in Fig. 6b to analyze the accuracy of different methods. From Fig. 6b, we can see that the curve representing "ite" overlaps with that of the golden result. Besides, the three curves representing "lin", "lin & svd batch", and "lin & svd update"



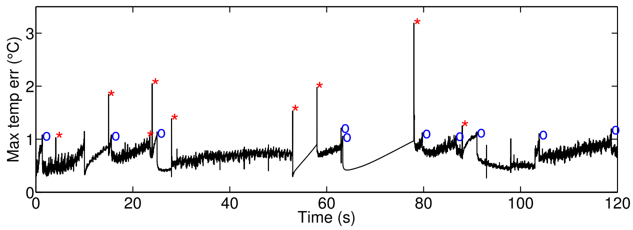
(a) Input dynamic power trace of the central grid of the chip.



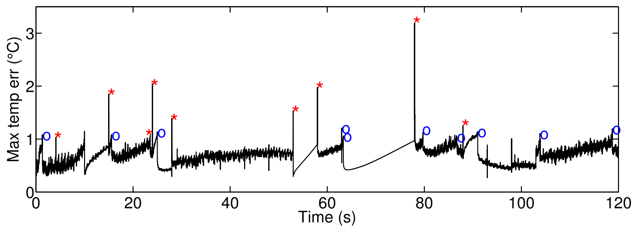
(b) The estimated temperature traces of the central grid by different methods for accuracy comparison.



(c) Temperature error trace of using "lin only" method. Blue circles represent updating full-sized thermal model with new Taylor expansion points .



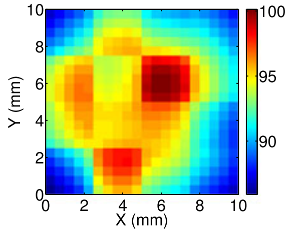
(d) Temperature error trace of using "lin & svd batch" method. Blue circles represent updating compact thermal model with former projector U_r, red stars represent updating both U_r and compact model when former projector becomes invalid.



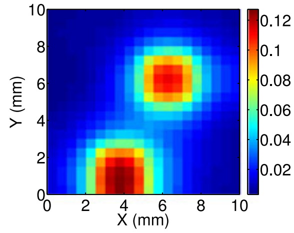
(e) Temperature error trace of using "lin & svd update" method. Blue circles represent updating compact thermal model with former projector U_r, red stars represent updating both U_r and compact model when former projector becomes invalid.

Fig. 6. Accuracy comparison and maximum estimation error traces of the proposed method on the 16-core chip.

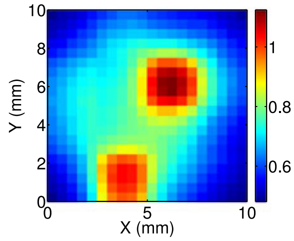
are very close to the golden curve as well, showing very small estimation errors. Such observation means that using Taylor expansion based local linear thermal models for



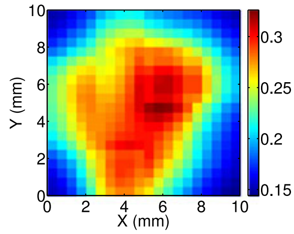
(a) Golden temperature distribution.



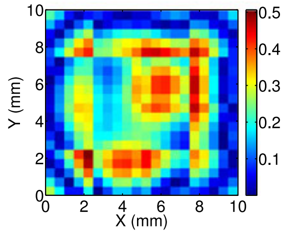
(b) Temperature error distribution of using iteration based method.



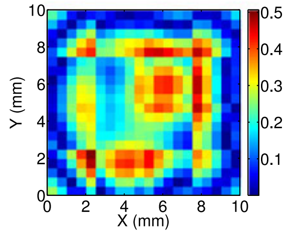
(c) Temperature error distribution of using TILTS method.



(d) Temperature error distribution of using "lin only" method.



(e) Temperature error distribution of using "lin & svd batch" method.



(f) Temperature error distribution of using "lin & svd update" method.

Fig. 7. Full-chip power distribution and thermal distribution estimation errors of the 16-core chip at a random time point.

thermal estimation is accurate (verified by curve "lin"), and further performing MOR on the linear thermal models (verified by curve "lin & svd batch") and even MOR with incremental SVD update (verified by curve "lin & svd update") introduces negligible estimation error. The curve of TILTS is far away from the golden one for most estimation time, showing large estimation error. This is because the linear leakage model of TILTS is only accurate around its single Taylor expansion point (80°C in this experiment).

We also would like to see how the new method changes linear models and updates the projection matrix U_r during the thermal estimation process. So, we plot the maximum transient temperature errors across the chip for all three cases in Fig. 6, and also mark the linear model change points and projection matrix U_r update points in the figure. We can see that for all three cases, every time the maximum thermal estimation error is going to violate our pre-defined error threshold, the linear thermal model is changed by using the new Taylor expansion points, resulting in an immediate and significant drop in estimation error, as expected. Furthermore, "lin & svd batch" and "lin & svd update" show slightly larger thermal estimation errors than "lin only" case. This means MOR did introduce small extra error as expected, but it can significantly reduce estimation time as will be shown later. The transient maximum thermal estimation error waveforms from "lin & svd batch" and "lin & svd update" are very similar to each other, meaning using

TABLE 2
Accuracy and Speed of Different Thermal Estimation Methods

core #	method	model size	temp err (°C) max	temp err (°C) avg	power err (%) max	power err (%) avg	time (s) max	time (s) avg	speedup vs ite min	speedup vs ite avg	speedup vs TILTS min	speedup vs TILTS avg
9	ite	912	3.84	0.07	6.53	0.12	58.59	56.22	NA		NA	
	TILTS		6.45	0.83	17.49	2.93	13.30	13.26	4.07	4.28	NA	
	lin only		3.96	0.17	7.52	0.19	8.06	7.67	6.85	7.33	1.64	1.72
	lin & svd batch	24	4.49	0.22	7.65	0.37	2.39	0.71	23.67	79.63	5.47	18.66
	lin & svd update		4.49	0.22	7.66	0.38	1.77	0.68	32.05	84.39	7.40	19.78
16	ite	1612	4.23	0.07	6.89	0.15	210.57	204.72	NA		NA	
	TILTS		7.05	0.81	16.96	2.72	42.79	41.47	4.81	4.94	NA	
	lin only		4.63	0.16	7.64	0.18	34.42	29.92	6.10	6.82	1.22	1.39
	lin & svd batch	32	4.76	0.19	7.74	0.32	12.29	1.79	16.27	112.57	3.35	22.70
	lin & svd update		4.76	0.19	7.74	0.32	5.09	1.15	39.48	174.54	8.11	35.19
25	ite	2512	3.44	0.03	4.88	0.05	466.70	457.55	NA		NA	
	TILTS		7.56	0.85	16.54	2.93	96.04	94.56	4.75	4.84	NA	
	lin only		4.60	0.10	5.55	0.19	61.91	61.70	7.31	7.42	1.51	1.53
	lin & svd batch	40	3.87	0.12	5.61	0.21	15.35	1.32	29.11	345.56	6.19	71.64
	lin & svd update		3.87	0.12	5.61	0.21	6.40	1.04	71.87	440.83	14.84	90.62
36	ite	3612	3.14	0.02	4.36	0.03	899.88	874.98	NA		NA	
	TILTS		6.17	0.79	14.73	2.83	194.97	193.62	4.29	4.52	NA	
	lin only		3.43	0.08	4.87	0.14	156.64	148.82	5.39	5.89	1.24	1.30
	lin & svd batch	50	3.59	0.11	5.14	0.19	39.31	2.61	22.39	336.69	4.83	72.80
	lin & svd update		3.59	0.11	5.14	0.19	13.17	1.49	66.82	589.49	14.43	127.28

Time is reported for 120s thermal estimation.

incremental SVD update does not introduce extra error by further boosting the estimation speed (will be shown later).

In addition to showing the transient results of the probe grid, we also plot the estimated full-chip temperature error distribution snapshot at a random time point in Fig. 7. It is clear that the results given by using iteration method (see Fig. 7b) are very accurate, with temperature error across the chip to be within 0.12°C. By using local linear thermal models, the error becomes a little larger, but still very small with the largest temperature error to be within 0.3°C, as shown in Fig. 7d. Then, by applying MOR (whether with batch SVD or further introducing the incremental SVD update technique) to the linear thermal models (see Figs. 7e and 7f), the error of "lin & svd update" method is almost identical to previous batch SVD one, meaning that the incremental SVD update technique does not introduce additional thermal estimation error. Last, the error of TILTS (see Fig. 7c) is much larger because it can only use one Taylor expansion point.

5.3 Speed and Accuracy Data of the Proposed Method

We have graphically seen from Section 5.2 that the new method has good accuracy. Now in this part, we show the speed and accuracy comparison results of the new method against the iterative method and TILTS, when they are applied on different multi-core systems. For each multi-core system, we generate 100 different dynamic power traces with different combinations of SPEC benchmark power traces and scale patterns, then perform transient thermal simulations using different methods on all power traces.

Table 2 records the speed and accuracy data of the new method comparing with "ite" and TILTS. For estimation accuracy, "ite" performs best as expected. This is because "ite" uses the full-sized thermal model and the leakage's nonlinear effect is handled accurately at each time step through iteration. "lin only" gives slightly larger errors than "ite", because it is based on the linear approximation of static power at several Taylor expansion points. Errors given by "lin & svd batch" method are slightly larger than those of the "lin only" one, simply because it uses a reduced thermal model. Accuracy performance of "lin & svd update" is almost the same as "lin & svd batch", from which we can conclude that performing incremental SVD update on the projection matrix U_r introduces negligible error. We also show frequency response comparison in Fig. 8. In the figure, we plot a diagonal element (denote as $h(s)$) of the transfer function matrix $H(s) = L(G_l + sC)^{-1}B$. Since $h(s)$ is a diagonal element of the transfer function matrix, it represents the transfer function of input (power) and output (temperature) at the same thermal node on chip. The transfer functions of the reduced models for the same input and output pair are also plotted. We can see that the frequency responses of the

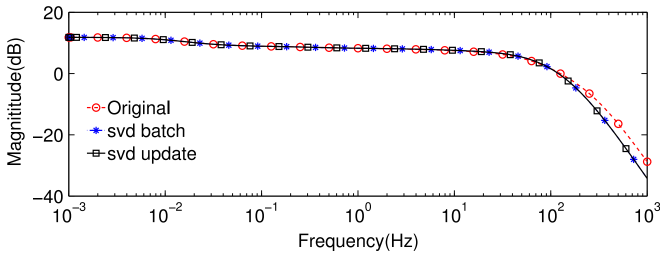


Fig. 8. Frequency responses of 16-core system thermal models, with input (power) and output (temperature) at the same thermal node on chip. "Original" represents original full-size model, "svd batch" and "svd update" represent reduced models after MOR process with pure SVD and incremental SVD, respectively.

TABLE 3
Detailed Computing Time Analysis with the 16-Core System Case

method	model update				U_r update				analysis time (s)		total time (s)	
	count		time (s)		count		time (s)					
	max	avg	max	avg	max	avg	max	avg	max	avg	max	avg
lin only	20	7.33	7.26	2.66	NA				28.10	27.26	34.42	29.92
lin & svd batch	32	17.07	0.23	0.11	14	1.12	11.28	0.99	0.78	0.69	12.29	1.79
lin & svd update	32	17.00	0.23	0.11	14	1.12	4.27	0.36	0.71	0.68	5.09	1.15

"Model update" represents thermal model update, "U_r update" means re-generating U_r, "analysis time" stands for pure transient simulation time.

reduced models given by "lin & svd batch" and "lin & svd update" overlap each other, showing high accuracy of the reduction. Finally, TILTS has the worst accuracy performance of all methods. This is because TILTS can only have a single Taylor expansion point (in the experiment, at 80°C), and is only accurate around that expansion point. TILTS does have smaller integration-based truncation error than the proposed method (in the experiment, truncation error of TILTS equals to estimation with 0.0001 s time step, while that of the proposed method equals to estimation with 0.01 s time step). However, for leakage-aware thermal estimation, error caused by leakage linearization can be much larger than integration-based truncation errors. As a result, TILTS has larger final thermal estimation error than the proposed method, because of its large leakage linearization error.

Now let us look at the speed comparison in Table 2. First, TILTS and "lin only" are several times faster than "ite" because they both use linear thermal model to avoid the time-consuming iteration process. It is noted that "lin only" is a little faster than TILTS, which is explained as follows. The essence of TILTS is to perform thermal estimation using time step equals to power sampling interval to achieve a faster speed, but still keeps integration-based truncation error as the same as HotSpot with a smaller time step. In our experiment, time steps of all methods are set to be the power sampling interval (0.01 s) as practical settings for runtime thermal estimation to ensure estimation speed. In this case, TILTS show no advantage in speed as expected, because it has no time step advantage. It is even a little slower than the proposed method, because the proposed method uses forward and backward substitutions with pre-factorized *sparse* LU decomposition, which is almost linear for sparse matrices, while TILTS needs to compute the matrices from the traditional simulation method first and perform *dense* matrix-vector multiplications, with higher complexity $O(n^2)$.

"lin & svd batch" shows even better speedup than "lin only" method, benefiting from using the smaller reduced thermal model. "lin & svd update" performs best in estimation speed. It uses the same sized reduced thermal model as "lin & svd batch" method, but it is much faster. This is because "lin & svd update" employs incremental SVD update to generate the new projection matrix U_r, which greatly reduces the overhead of the MOR process.

In order to see more detailed computing time information other than the total time, we perform an in-depth analysis with the 16-core system case. We divide the total computing time into three components: "model update", "U_r update", and "analysis". "model update" represents updating the thermal model when the Taylor expansion points become invalid (also count in MOR process using *previous* projection matrix U_r for "lin & svd batch" and "lin & svd update" cases). "U_r update" means re-generating the projection matrix U_r in MOR when it becomes invalid. "analysis time" stands for pure transient simulation time using thermal models, excluding the thermal model change time (counted in "model update") and projection matrix re-computing time (counted in "U_r update"). The results are show in Table 3. We can see that "lin & svd batch" and "lin & svd update" are much faster than "lin only" on both "model update" and "analysis time" parts. The speedup from "analysis time" is obviously due to the fact that the former two both use reduced thermal models in transient simulation. While the "model update" speedup is simply because the LU decompositions of the former two are based on the reduced thermal model matrices. For "U_r update" time, "lin & svd update" shows great advantage than "lin & svd batch". Such benefit is gained by the incremental SVD update technique introduced in Section 4.4.3. For "lin & svd batch" and "lin & svd update", their "U_r update" time will be significant when the update count is large, leading to a slow down in estimation speed. This extreme case happens when the chip temperature changes frequently and drastically in the estimation duration. But even for such worst case in our 100 tests, "lin & svd update" still gains around $39\times$ and $8\times$ speedup against "ite" and TILTS, respectively, as shown in Table 2.

5.4 Sensitivity Analysis of Taylor Expansion Temperature Region Length

In this section, we analyze the impact of Taylor expansion temperature region length on the accuracy and speed of the new thermal estimation method.

We perform thermal estimation using "lin & svd update" on the 16-core system, with different temperature region lengths ranging from 1°C to 80°C. The accuracy and computing cost results are shown in Fig. 9. We see that the estimation error is smaller with shorter region length, because more updates of thermal model and projection matrix are performed. However, if the region length is shorter than 10°C, decreasing region length further has negligible contribution in estimation accuracy, meaning that each linear approximation of leakage current can approximately cover around 10°C temperature range. On the computing time side, as shown in Fig. 9, very short temperature region length leads to large computing time cost, especially when the length is shorter than 10°C, because a lot of thermal model and projection matrix updates are performed. When the region length increases from 10°C to 40°C, there is a small time cost increase, because larger region length results in lager estimation error and leads to a little more updates. Lastly, when the temperature region length becomes very

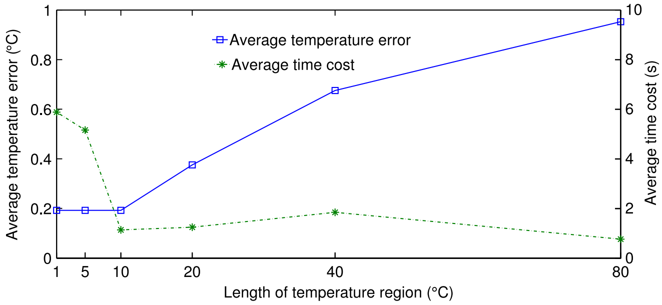


Fig. 9. Accuracy and computing cost analysis of using different temperature region lengths, with “lin & svd update” method on the 16-core chip.

long (for example, around 80°C), the time cost will decrease again, because node temperature hardly gets out of this long range to trigger updates. However, since the thermal estimation accuracy will be poor for such long region length, it is a bad choice in general unless extremely fast speed is required. In general, we see that the temperature region length affects the trade-off between estimation accuracy and computing cost, so it needs to be chosen according to the desired balance point of accuracy and cost.

6 Conclusion

In this article, we have demonstrated a new fast full-chip transient thermal estimation method. The new method uses Taylor expansion based local linearization technique to avoid the time-consuming iterations used in the traditional thermal analysis methods. A new linear thermal model is also formulated for easy transient simulation of temperature and static power. In order to further increase the thermal estimation speed, a specially designed model order reduction method with partial and incremental SVD update technique has been integrated into the estimation framework to generate local compact thermal models. The new method has been tested on several multi-core chips with SPEC benchmarks. The results show that the new method is able to accurately estimate full-chip transient temperature distribution. On different chips with core number ranging from 9 to 36, it achieved 85× to 589× speedup in average against traditional iteration based method, with average thermal estimation error to be around 0.2°C.

Acknowledgments

This research is supported in part by National Natural Science Foundation of China under grant No. 61404024, in part by the Fundamental Research Funds for the Central Universities under grant No. ZYGX2016J043, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

References

[1] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, “Dark silicon and the end of multicore scaling,” *IEEE Micro*, vol. 32, no. 3, pp. 122–134, May 2012.

[2] M. Taylor, “A landscape of the new dark silicon design regime,” *IEEE Micro*, vol. 33, no. 5, pp. 8–19, Oct. 2013.

[3] D. Brooks and M. Martonosi, “Dynamic thermal management for high-performance microprocessors,” in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit.*, Jan. 2001, pp. 171–182.

[4] V. Hanumaiah and S. Vrudhula, “Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling,” *IEEE Trans. Comput.*, vol. 63, no. 2, pp. 349–360, Feb. 2014.

[5] Z. Liu, S. X.-D. Tan, X. Huang, and H. Wang, “Task migrations for distributed thermal management considering transient effects,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 2, pp. 397–401, Feb. 2015.

[6] H. Wang, et al., “Hierarchical dynamic thermal management method for high-performance many-core microprocessors,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 22, no. 1, pp. 1:1–1:21, Jul. 2016.

[7] Y. Yang, Z. P. Gu, C. Zhu, R. P. Dick, and L. Shang, “ISAC: Integrated space and time adaptive chip-package thermal analysis,” *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 16, no. 1, pp. 86–99, Jan. 2007.

[8] H. Wang, S. X.-D. Tan, G. Liao, R. Quintanilla, and A. Gupta, “Full-chip runtime error-tolerant thermal estimation and prediction for practical thermal management,” in *Proc. Int. Conf. Comput. Aided Des.*, Nov. 2011, pp. 716–723.

[9] L. He, W. Liao, and M. R. Stan, “System level leakage reduction considering the interdependence of temperature and leakage,” in *Proc. Des. Autom. Conf.*, 2004, pp. 12–17.

[10] W. Huang, K. Sankaranarayanan, K. Skadron, R. J. Ribando, and M. R. Stan, “Accurate, pre-RTL temperature-aware processor design using a parameterized, geometric thermal model,” *IEEE Trans. Comput.*, vol. 57, no. 9, pp. 1277–1288, Sep. 2008.

[11] S. R. Sarangi, G. Ananthanarayanan, and M. Balakrishnan, “LightSim: A leakage aware ultrafast temperature simulator,” in *Proc. Asia South Pacific Des. Autom. Conf.*, 2014, pp. 855–860.

[12] P. Liu, et al., “Fast thermal simulation for architecture level dynamic thermal management,” in *Proc. Int. Conf. Comput. Aided Des.*, 2005, pp. 638–643.

[13] L. Codecasa, D. D’Amore, and P. Maffezzoni, “Boundary condition independent compact models of dynamic thermal networks with many heat sources,” in *Proc. Therm. Thermomech. Phenom. Electron. Syst.*, 2006, pp. 685–689.

[14] H. Wang, S. X.-D. Tan, D. Li, A. Gupta, and Y. Yuan, “Composable thermal modeling and simulation for architecture-level thermal designs of multi-core microprocessors,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 18, no. 2, pp. 28:1–28:27, Mar. 2013.

[15] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, “HotSpot: A compact thermal modeling methodology for early-stage VLSI design,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 14, no. 5, pp. 501–513, May 2006.

[16] E. Rotem, A. Naveh, D. Rajwan, A. Ananthakrishnan, and E. Weissmann, “TILTS: A fast architectural-level transient thermal simulation method,” *J. Low Power Electron.*, vol. 3, no. 1, pp. 13–21, Apr. 2007.

[17] Y. Zhan and S. S. Sapatnekar, “A high efficiency full-chip thermal simulation algorithm,” in *Proc. Int. Conf. Comput. Aided Des.*, 2005, pp. 634–637.

[18] J.-H. Park, A. Shakouri, and S.-M. Kang, “Fast evaluation method for transient hot spots in VLSI ICs in packages,” in *Proc. Int. Symp. Quality Electron. Des.*, 2008, pp. 600–603.

[19] Y. Liu, R. Dick, L. Shang, and H. Yang, “Accurate temperature-dependent integrated circuit leakage power estimation is easy,” in *Proc. Eur. Des. Test Conf.*, 2007, pp. 1–6.

[20] H. Sultan and S. R. Sarangi, “A fast leakage aware thermal simulator for 3D chips,” in *Proc. Eur. Des. and Test Conf.*, Mar. 2017, pp. 1733–1738.

[21] W. Wu, L. Jin, J. Yang, P. Liu, and S. Tan, “A systematic method for functional unit power estimation in microprocessors,” in *Proc. Des. Autom. Conf.*, Jun 2006, pp. 554–557.

[22] M. Powell, A. Biswas, J. Emer, S. Mukherjee, B. Sheikh, and S. Yardi, “CAMP: A technique to estimate per-structure power at run-time using a few simple parameters,” in *Proc. IEEE Int. Symp. High-Perform. Comput. Archit.*, Feb. 2009, pp. 289–300.

[23] H. Wang, S. X.-D. Tan, X.-X. Liu, and A. Gupta, “Runtime power estimator calibration for high-performance microprocessors,” in *Proc. Eur. Des. Test Conf.*, Mar. 2012, pp. 352–357.

[24] A. Abdollahi, F. Fallah, and M. Pedram, “Leakage current reduction in CMOS VLSI circuits by input vector control,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 12, no. 2, pp. 140–154, Feb. 2004.

[25] R. Shen, S. X.-D. Tan, H. Wang, and J. Xiong, “Fast statistical full-chip leakage analysis for nanometer VLSI systems,” *ACM Trans. Des. Autom. Electron. Syst.*, vol. 17, no. 4, pp. 51:1–51:19, Oct. 2012.

[26] W. Liu, K. Cao, X. Jin, and C. Hu, “BSIM 4.0.0 technical notes,” EECS Dept., Univ. California, Berkeley, Ca, Tech. Rep. UCB/ERL M00/39, 2000. [Online]. Available: http://www.eecs.berkeley.edu/Pubs/TechRpts/2000/3863.html

[27] J. C. Ku, S. Ozdemir, G. Memik, and Y. Ismail, "Thermal management of on-chip caches through power density minimization," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 15, no. 5, pp. 592–604, May 2007.
[28] A. C. Antoulas, *Approximation of Large-Scale Dynamical Systems.* Philadelphia, PA, USA: The Society for Industrial and Applied Mathematics (SIAM), 2005.
[29] S. X.-D. Tan and L. He, *Advanced Model Order Reduction Techniques in VLSI Design*. Cambridge, U.K.: Cambridge Univ. Press, 2007.
[30] H. Wang, S. X.-D. Tan, and R. Rakib, "Compact modeling of interconnect circuits over wide frequency band by adaptive complex-valued sampling method," *ACM Trans. Des. Autom. Electron. Syst.*, vol. 17, no. 1, pp. 5:1–5:22, Jan. 2012.
[31] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Proc. Eur. Conf. Comput. Vis.*, May 2002, pp. 707–720.
[32] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: A framework for architectural-level power analysis and optimizations," in *Proc. Int. Symp. Comput. Archit.*, 2000, pp. 83–94.

Hai Wang received the BS degree from Huazhong University of Science and Technology, China, and the MS and PhD degrees from University of California, Riverside, in 2007, 2008, and 2012, respectively. He is currently an associate professor with the University of Electronic Science and Technology of China. His research interests mainly lie in electrical/thermal verification and optimization of VLSI circuits and systems. He has served as technical program committee member of several international conferences including DATE, ASP-DAC and ISQED, and also served as reviewer of many journals including *IEEE Transactions on Computers*, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, and *ACM Transactions on Design Automation of Electronic Systems.*

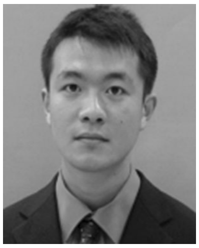
Jiachun Wan received the bachelor's degree in microelectronics from the University of Electronic Science and technology of China, in 2015. Currently, he is working toward the master's degree with UESTC. His current research interests include thermal analysis, power analysis, and thermal management of integrated circuit.

Sheldon X.-D. Tan (S'96-M'99-SM'06) received the BS and MS degrees in electrical engineering from Fudan University, Shanghai, China, and the PhD degree in electrical and computer engineering from the University of Iowa, Iowa City, in 1992, 1995 and 1999 respectively. He is an associate director of computer engineering programn (CEN) and is a professor in the Department of Electrical Engineering, University of California, Riverside, CA. His research interests include VLSI reliability modeling, optimization and management at circuit and system levels, thermal modeling, optimization and dynamic thermal management for many-core processors, statistical modeling, simulation and optimization of mixed-signal/RF/analog circuits, parallel circuit simulation techniques based on GPU and multicore systems. He received NSF CAREER Award in 2004 and received Outstanding Oversea Investigator Award from the National Natural Science Foundation of China (NSFC) in 2008. He received the Best Paper Award from 2007 IEEE International Conference on Computer Design (ICCD'07), the Best Paper Award from 1999 IEEE/ACM Design Automation Conference. He also receives three Best Paper Award Nomination from IEEE/ACM Design Automation Conferences in 2005, 2009 and 2014 and one Best Paper Award nomination from ASP-DAC in 2015. He now is serving as the editor-in-chief of the *Integration, The VLSI Journal*. He is also serving as an associate editor of the three journals: *IEEE Transactions on VLSI Systems (TVLSI)*, the *ACM Transactions on Design Automation of Electronic Systems (TODAES)*. He is a senior member of the IEEE.

Chi Zhang received the bachelor's degree from Taiyuan University of Science and Technology, and the master's degree from the Microelectronics Research Institute of Chinese Academy of Sciences, in 1994 and 2003. He is currently working toward the PhD degree with the University of Electronic Science and technology of China. His main research directions are mixed-signal integrated circuit design, EDA technology, multimode biometrics technology.

He Tang (M'09) received the BSEE degree from the University of Electronic Science and Technology of China, Chengdu, China, the MS degree in electrical and computer engineering from the Illinois Institute of Technology, Chicago, and the PhD degree in electrical engineering from University of California, Riverside, in 2005, 2007, and 2010. From 2010 to 2012, he was with OmniVision Technologies Inc., in Santa Clara, California, as an Analog IC Designer, where he worked on high-speed I/O inteface. Since 2012, he has been an associate professor and subsequently a professor with the University of Electronic Science and Technology of China, Chengdu, China. He has authored or coauthored more than 40 papers. His research interests focus on data converters and analog/mixed-signal IC designs. His pas work includes high-speed high-resolution pipelined ADCs with digital calibration and high-performance ultra-low-power SAR ADCs. He has served on IEEE CAS Analog Signal Processing Technical Committee (ASPTC) since 2013. He is a member of the IEEE.

Yuan Yuan received his BS and MS degrees from the University of Electronic Science and Technology of China, in 1992 and 2005, respectively. He is currently an associate professor with the University of Electronic Science and technology of China. His main research directions are electronic measuring equipment design, computer based measuring technology, embedded system, etc. He has published more than 10 research papers in international conferences and journals.



Keheng Huang received the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. He is a senior engineer of Southwest China Research Institute of Electronic Equipment. His current research interests include digital signal processing and array design.

Zhenghong Zhang received the BEng and MEng degrees in electronic engineering from Xidian University, Xi'an, China. He is currently a chief engineer of Southwest China Research Institute of Electronic Equipment. His current research interests include digital signal processing, microsystem, and artificial intelligence.