單晶片系統驗證之核心技術開發

子計畫六：針對先進晶片設計的熱點驗證之完整熱模型與高效能熱分析(3/3)

Compact Thermal Modeling and Efficient Thermal Simulation for Hot Spots Verifications of Modern IC Designs

計畫編號：NSC 96－2220－E009－012

執行期間：96 年 8 月 1 日 至 97 年 7 月 31 日

計畫主持人：李育民

一、中文摘要

在現今的積體電路設計，能夠精確預測電路溫度分佈對於電路時序分析(timing analysis)、減少漏電流、消耗功率評估、熱點的避免和可靠度分析是相當重要的。本計劃主要目的為發展晶片上預測電路溫度分佈的快速分析工具。

在計劃的前兩年中，我們已利用一般化的積分轉換(generalized integral transforms)技術針對設計自動化流程前端發展出一套有效率的熱分析工具，並應用此一技術分析 3D 積體電路的熱分佈[R1~R3]；同時亦針對現今在漏電流主導的製程技術下，規劃了一個統計型的晶片溫度分佈分析的初步流程。

在計劃的最後一年，我們將完整的規劃此一流程、實現分析的方法並驗證其準確度與效率。

關鍵詞：一般化的積分轉換；製程變異；熱分析；溫度剖面；熱點；漏電流

二、英文摘要

The capability of predicting the temperature profile is critically important for circuit timing analysis, leakage reduction, power estimation, hotspots avoidance, and reliability concerns during modern IC designs. This work presents an accurate and fast analytical full-chip thermal simulator for the temperature-aware chip design.

In the previous two years, we have developed a generalized integral transforms method to solve the transient and steady temperature distribution for the thermal placement stage, and extended the proposed GIT based method to deal with 3D ICs thermal simulation[R1~R3].

In this year, we have developed a stochastic thermal simulation procedure with considering the leakage power variation because of the effects of process variations, implemented the proposed method, and demonstrated its accuracy and efficiency.

Keywords：Generalized Integral Transforms, Process Variations, Spatial Correlation, Thermal Analysis, Temperature Profile, Hotspot, Leakage Current, Leakage Power

三、研究計畫之背景及目的

Because of the drastically increasing power consumption of integrated circuits, the thermal issue has become one of the most important concerns in VLSI design. The high temperature distribution and thermal gradient variation have serious impacts on the timing, power and reliability of designs [2]–[8].

Conventional thermal simulators [2]–[8] are only conducted by solving the heat transfer equations with the nominal power consumption of the die. However, as the technology scales down, the decreased controllability of processes have caused considerable variations of leakage power [1]. The variations of leakage power are expected as high as 20 times caused by 30% within-die process variations [1], and the related fluctuations of temperature distribution are significant. Those unreliably optimistic estimations [2]–[8] might guide designers to the wrong design direction and lead to low yields. On the contrary, the deterministic simulation with the worst-case parameters can result in the immoderate guard-banding and can cause low performance [11]. These undesirable phenomena lead the statistical thermal simulation to be essential, especially for the leakage power dominated technology.

Thermal simulations can be generally divided into two categories as transient-analysis and steady-state analysis. Transient-analysis is concerned with the evolution of temperature distribution within a chip given a time-varying power density distribution. As indicated in [2], [6], [13], the thermal time constant of heat conduction is much larger than the clock period of circuit. This fact leads to steady-state thermal analysis is more interested to study the stability of temperature distribution with a given power density distribution averaged over time.

In this work, we will focus on the steady-state thermal analysis with considering withindie process variations with spatial correlation. Although we do not consider electro-thermal coupling due to the scope of this paper, our simulator can be readily combined with the temperature-dependent electrical modeling to perform an iteratively update scheme, for example, consider the temperature-dependent subthreshold leakage power in the thermal simulation.

By using KL expansion [12], we transform the physical parameters with variations to a set of uncorrelated random variables and employ the PCs scheme [12] and stochastic Galerkin procedure to convert the stochastic thermal problem to a set of deterministic problems. After that, any existing deterministic thermal simulator such as [2]–[8] can be used to solve those deterministic heat transfer equations. Finally, the mean and variance profiles of the steady-state temperature for the full chip can be evaluated.

Our major contributions are

1) To the authors' best knowledge, this is the first stochastic thermal simulator considering within-die process variations with spatial correlation. We also demonstrate that the deterministic simulators with nominal physical parameters underestimate the temperature distribution, and are unreliable in the nanometer technology.

2) Our simulator can accurately and efficiently provide the mean and standard deviation profiles of the temperature to guide designers avoiding thermal failures due to process variations.

3) Experimental results indicate that ignoring process variations with spatial correlation during the thermal simulation is not allowable, and can induce several issues of design and reliability. The rest of this report is organized as follows. The problem formulation is introduced in section 四、A. The flowchart of proposed stochastic thermal simulation is

presented in section 四、B. Then, the modeling of physical parameters are described in section 四、C, the leakage power modeling is illustrated in section 四、D, and the stochastic Galerkin procedure is addressed in section 四、E. The experimental results are given in section 四、F. Finally, the conclusion and achievements are given in sections 五 and 六, respectively.

四、研究方法

A. Problem Formulation

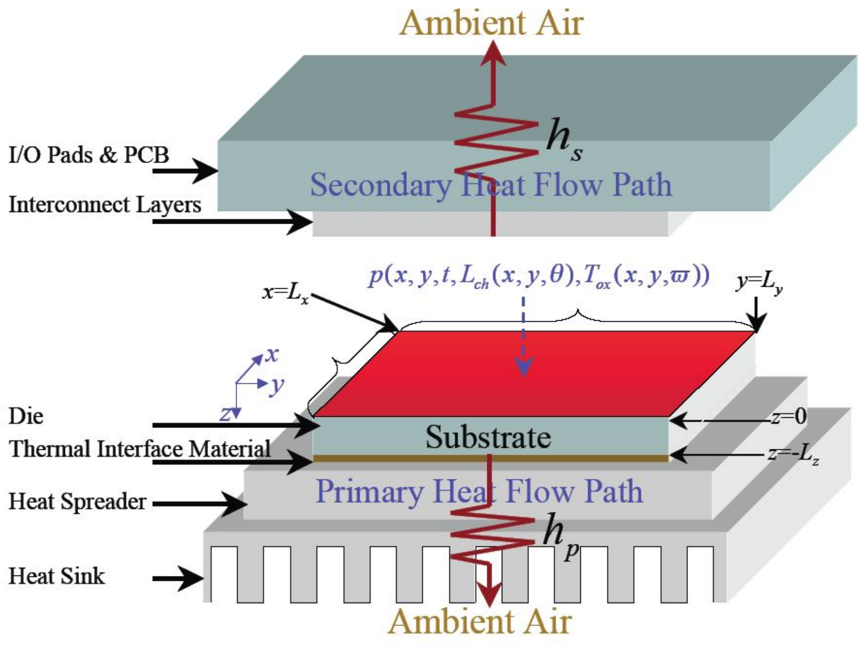


Fig. 1. Compact thermal model of physical design.

The typical compact thermal model for physical design stages [2]–[8] is shown in Fig. 1. It consists of three portions. The primary heat flow path is composed of thermal interface material, heat spreader and heat sink. The secondary heat flow path contains interconnect layers, I/O pads and the print circuit board. The functional blocks on the die are modeled as many power generating sources attached to the thin layer close to the top surface of the die with the thickness being equal to the junction depth of device [9]. The devices consuming the dynamic and leakage power are the mainly heat sources. Generally, the dynamic power is insensitive to process variations and can be assumed to be deterministic [10]. The leakage power is greatly affected by physical parameters such as the channel length and oxide thickness, and needs to be treated as spatial random processes [11]. By combining the compact thermal model and statistical power dissipation, the steady state temperature bT(r; _;$) of die is governed by the following stochastic steady-state heat transfer equation.

$$\nabla\cdot\left(\kappa\left(\mathrm{r},\hat{T}\right)\nabla\hat{T}\left(\mathrm{r},\theta,\varpi\right)\right)=-p\left(\mathrm{r},L_{ch}\left(x,y,\theta\right),T_{ox}\left(x,y,\varpi\right)\right), \qquad (1)$$

subject to the following boundary condition

$$\kappa\left(\mathrm{r}_{b_s},\hat{T}\right)\frac{\partial\hat{T}\left(\mathrm{r}_{b_s},\theta,\varpi\right)}{\partial n_{b_s}}+h_{b_s}\hat{T}\left(\mathrm{r}_{b_s},\theta,\varpi\right)=f_{b_s}\left(\mathrm{r}_{b_s}\right), \tag{2}$$

where $\mathrm{r}=(x,\, y,\, z)\in D$, $D=(0,L_x)\times(0,L_y)\times(-L_z,0)$ is the domain of die, L_x and L_y are lateral sizes of die, L_z is the thickness of die, $\kappa\left(\mathrm{r},\, \hat{T}\right)$ is the thermal conductivity $\left(W/m\cdot{}^{\circ}C\right)$ of die, ∇ is the diverge operator, b_s is any specific boundary surface of the die, r_{bs} is the position located on bs, hbs is the heat-transfer coefficient on b_s, h_{b_s} is the heat flux function on b_s, $\partial/\partial n_{b_s}$ is the differentiation along the outward direction normal to b_s, θ and ϖ are sampling values of manufacturing outcomes $\Omega_{L_{ch}}$ and $\Omega_{T_{ox}}$ for the channel length and oxide thickness, respectively, $L_{ch}\left(x,y,\theta\right)$ and $T_{ox}\left(x,y,\varpi\right)$ are the random processes of the device channel length and the oxide thickness, respectively, $p\left(\mathrm{r},L_{ch}\left(x,y,\theta\right),T_x\left(x,y,\varpi\right)\right)$ is the random process of power density profile which consists of dynamic power density profile $p_d\left(\mathrm{r}\right)$, subthreshold leakage power density profile $p_s\left(\mathrm{r},L_{ch}\left(x,y,\theta\right)\right)$, and gate leakage power density profile $p_g\left(\mathrm{r},T_x\left(x,y,\varpi\right)\right)$. Since the major part of device current passes through the channel, the power density distribution has its value only when $\mathrm{r}\in\left(0,L_x\right)\times\left(0,L_y\right)\times\left(-j_d,0\right)$. Here, j_d is the junction depth of device [9].

Generally, the values of $\kappa\left(\mathrm{r},\, \hat{T}\right)$ are temperature dependent. For the deterministic thermal simulation, the difference of peak temperature is about $5^{\circ}C$ between the result with temperature-dependent thermal parameters and the result with constant thermal parameters at $25^{\circ}C$ [5]. In current VLSI design, the on-die temperature can be in the degree of $100^{\circ}C$. Under this situation, this difference may lead to about 5% error for the peak temperature of die. Since the effort to amend this error is relatively high1, for practical purposes, these thermal parameters are usually treated as appropriate constants while performing temperature aware floor-planning and placement [13].

In this work, the reasonable value of each thermal parameter is set at the roughly steady-state average mean temperature of die which is got by using an iteratively computational scheme to the simplified 1-D thermal model shown in Fig. 2. Please see section 四、F for the detail of using 1-D thermal model. By using these estimated thermal parameters, the error of peak steady-state temperature can be reduced.

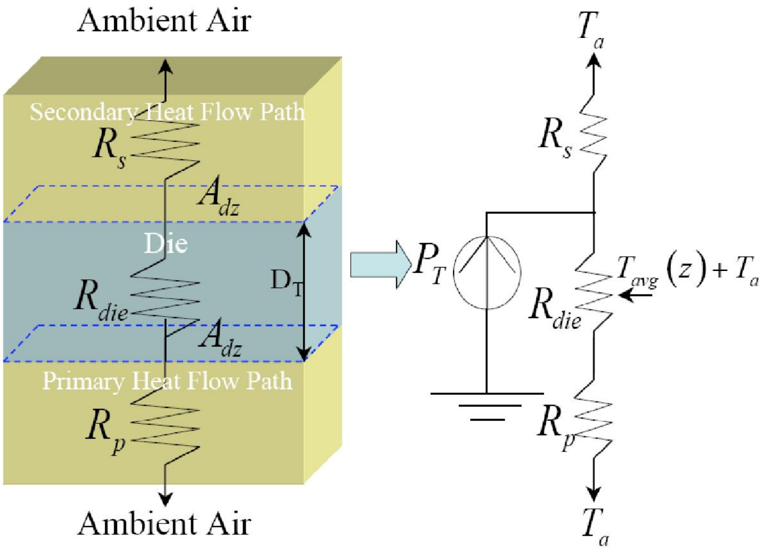


Fig. 2. Simplified 1-D thermal model for obtaining the roughly average rising mean temperature of die. The modeled thermal resistance network is shown in the right hand side. The values of thermal resistors are $R_s = 1/A_{dz}h_s$, $R_p = 1/A_{dz}h_p$ and $R_{die} = D_T/\kappa A_{dz}$. $T_{avg}(z)$ is the average rising mean temperature with respect to the room temperature T_a of lateral planes at arbitrary z position of the die. Here, R_{die} can be viewed as a variable resistor when obtaining $T_{avg}(z)$ at certain z position. P_T is the total mean power consummation of the die. A_{dz} is the cross area of die normal to the z-direction and D_T is the thickness of the die.

With the above description, the stochastic heat transfer equations for the steady state rising temperature $T(\mathrm{r},\theta,\varpi) = \hat{T}(\mathrm{r},\theta,\varpi) - T_a$ of die can be written as

$$\kappa\nabla^2 T(\mathrm{r},\theta,\varpi) = -p\big(\mathrm{r}, L_{ch}(x,y,\theta), T_{ox}(x,y,\varpi)\big), \tag{3}$$

subject to the boundary condition

$$\kappa\frac{\partial T(\mathrm{r}_{b_s},\theta,\varpi)}{\partial n_{b_s}} + h_{b_s}T(\mathrm{r}_{\mathrm{b}_s},\theta,\varpi) = \hat{f}_{\mathrm{b}_s}(\mathrm{r}_{\mathrm{b}_s}), \tag{4}$$

where κ is the thermal conductivity of die got by using the roughly steady-state average mean temperature, $\hat{f}_{bs}(\mathrm{r}_{b_S})$ is a modified heat flux function on b_s , and T_a is the ambient temperature.

With the above stochastic steady-state heat transfer equations, we are going to evaluate the mean and variance profiles of the steady state full-chip temperature.

B. Stochastic thermal simulation flowchart

The executing flow of the proposed stochastic thermal simulator is summarized in Fig. 3. Given the spatial covariance functions of physical parameters, we transfer spatially the correlated physical parameters such as the channel length and the oxide thickness into a set of uncorrelated random variables by using the KL expansion. With these uncorrelated random variables, the PCs are built to serve as polynomial bases for approximating the die temperature. Then, the leakage current models for different types of gates are built for modeling the power of gates by applying the minimal least square fitting to the simulation results of HSPICE under the TSMC 65 nm technology.

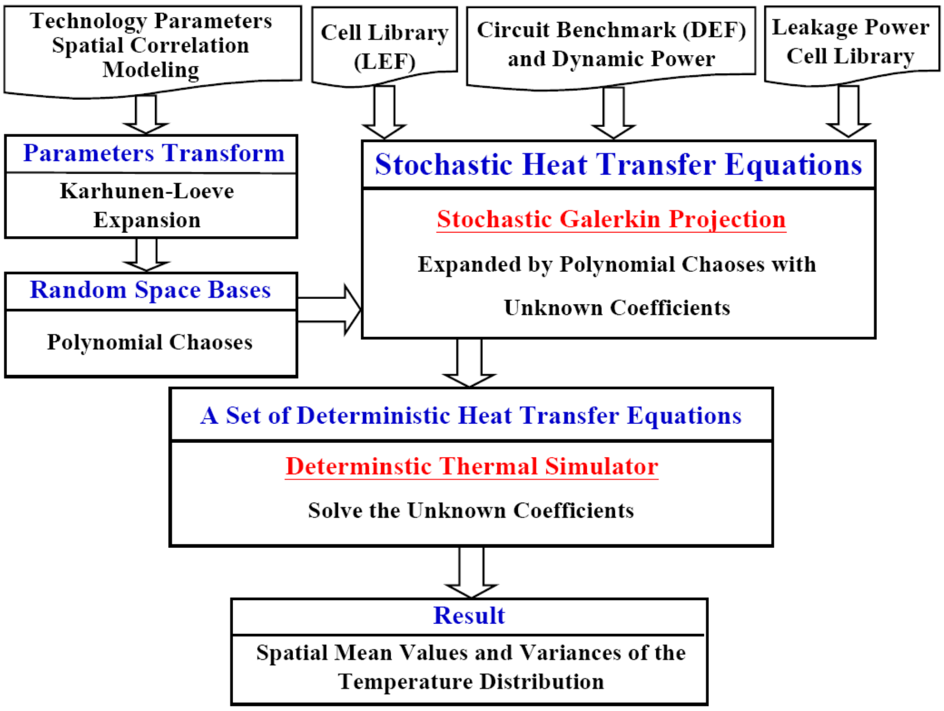


Fig. 3. Stochastic thermal simulation flowchart.

After the chip geometry, the gate level placement, the package configuration, the dynamic and the stochastically leakage power density profiles are obtained, the compact thermal model shown in Fig. 1 is constructed. Then, the stochastic Galerkin projection method [12] is employed to convert the stochastic heat transfer equations to a set of deterministic heat transfer equations. The number of these deterministic heat transfer equations is equal to the number of total PCs. Finally, an efficiently deterministic thermal simulator [8] is utilized to solve these deterministic heat transfer equations, and the mean and variance profiles for the steady-state full-chip temperature are obtained.

C. Parameter modeling

The number of random variables and the computational complexity of simulation severely increase when considering the spatial correlation of within-die process variations. A well known technique to reduce the above difficulties is the Principal Component Analysis (PCA) [14], which is a grid-based method. However, the nature of PCA has the limitation of high-dimensional parameter modeling. An alternative formulation without the drawback of grid-based methods for tackling with the correlated parameters is the KL expansion. With the same level of accuracy, the number of random variables used by the KL expansion is significantly smaller than the PCA’s [15].

(1) Karhunen-Loeve expansion

The KL expansion [19] of a second-order random process $\alpha(x,y,\vartheta)$ with a continuous spatial covariance function is

$$\alpha\left(x,y,\vartheta\right)=\alpha\left(x,y\right)+\sum_{k=1}^{\infty}\sqrt{\gamma_k}\phi_k\left(x,y\right)\eta_k\left(\vartheta\right), \tag{5}$$

where $\bar{\alpha}(x,y)$ is the mean of $\alpha(x,y,\vartheta)$, and each γ_k and each $\phi_k(x,y)$ are the eigenvalue and the eigenfunction derived from the following Fredholm integral equation.

$$\int_{Do} C\left(x_1,x_2\right)\phi_k\left(x_2\right)dx_2=\gamma_k\phi_k\left(x_1\right). \tag{6}$$

Here, $C(\mathrm{x}_1,\mathrm{x}_2)$ is the covariance function of the random process $\alpha(x,y,\vartheta)$, $(x,y)\in D_0=(0,L_x)\times(0,L_y)$ is the plane at the top surface of die, $\mathrm{x}_1=(x_1,y_1)$, $\mathrm{x}_2=(x_2,y_2)$, ϑ is the sampling event of sample space Ω_α, and $\{\eta_k(\vartheta)\}$ is a set of uncorrelated random variables with each $\eta_k(\vartheta)$ being zero mean and unit variance.

The KL expansion satisfies the following properties [12] :

1) The minimized mean-square error property for a finite-term representation of a random process.

2) It is unique for a random process with a given covariance function.

3) $\{\eta_k(\vartheta)\}$ is a set of independent standard normal random variables if the target random process is Gaussian.

Since values of physical parameters such as the oxide thickness and channel length are bounded, they are second-order random processes [19]. Moreover, as indicated in [15], [20]–[22], continuous spatial correlation functions for physical parameters such as exponential, Gaussian, linear, or a fitting form from the experimental data are suggested to be used. Combining with the second-order property and the continuity of above covariance functions, practically, the KL expansions of physical parameters are valid.

(2) Spatial correlation modeling

As indicated in [20], the spatial covariance function is not monotonically decreasing as the distance increases because the decreasing rates of the spatial covariance in the x- and y-directions are different. In order to model the spatial covariance function with the above characteristic, we adopt the following spatial covariance function which was proposed by [15] instead of adopting the purely distance dependent spatial covariance function [21], [22].

$$C\left(\mathbf{x}_1,\mathbf{x}_2\right)=\sigma^2\exp\left(-\frac{|x_1-x_2|}{\eta_x}\right)\exp\left(-\frac{|y_1-y_2|}{\eta_y}\right), \tag{7}$$

where η_x and η_y are correlation lengths of the target random process in the x- and y-directions, respectively, σ is the standard deviation of the target random process.

Closed-form expressions of eigen-functions and eigen-values which satisfy equation (6) with $C(\mathrm{x}_1,\mathrm{x}_2)$ being stated in equation (7) can be found in [23]. Due to the limitation of space, expressions of eigenfunctions and eigen-values are not presented in this paper. With the closed-form expressions of the eigenvalues and eigenfunctions for the spatial covariance functions in equation (7), KL expansions of $L_{ch}(x,y,\theta)$ and $T_{ox}(x,y,\varpi)$ can be obtained as

$$L_{ch}\left(x,y,\theta\right)=\bar{L}_{ch}\left(x,y\right)+\sum_{n=1}^{N_{T_{ox}}}\sqrt{\chi_m}q_m\left(x,y\right)\xi_m\left(\varpi\right), \tag{8}$$

$$T_{ox}\left(x,y,\varpi\right)=\bar{T}_{ox}\left(x,y\right)+\sum_{n=1}^{N_{T_{ox}}}\sqrt{\beta_n}f_n\left(x,y\right)\varsigma_n\left(\varpi\right). \tag{9}$$

Here, $\bar{L}_{ch}(x,y)=E\{L_{ch}(x,y,\theta)\}$, $\bar{T}_{ox}(x,y)=E\{T_{ox}(x,y,\varpi)\}$, $f_n(x,y)$ and $q_m(x,y)$ are eigenfunctions of $T_{ox}(x,y,\varpi)$ and $L_{ch}(x,y,\theta)$, respectively, β_n and χ_m are eigenvalues of $T_{ox}(x,y,\varpi)$ and $L_{ch}(x,y,\theta)$, respectively, $\{\varsigma_n(\varpi)\}$ and $\{\zeta_n(\theta)\}$ are independently standard normal random variables because $T_{ox}(x,y,\varpi)$ and $L_{ch}(x,y,\theta)$ are physically similar to Gaussian process [20], and indeed, the random processes of oxide thickness and channel length are assumed to be independent. In this work, we employ the criterion $\gamma_{N+1}/\sum_{k=1}^{N+1}\gamma_k<\varepsilon$ with $\varepsilon=0.001$ to obtain the specified truncation numbers $N_{T_{ox}}$ and $N_{L_{ch}}$ for $T_{ox}(x,y,\varpi)$ and $L_{ch}(x,y,\theta)$, respectively. For the sake of notation simplicity, $\{\xi_n(\varpi,\theta)\}_{n=1}^{N_{KL}}$ is set as the union of $\{\varsigma_n(\varpi)\}_{n=1}^{N_{T_{ox}}}$ and $\{\zeta_n(\theta)\}_{n=1}^{N_{L_{ch}}}$, $N_{KL}=N_{T_{ox}}+N_{L_{ch}}$, and ξ_n, ς_n and ζ_n are used to represent $\xi_n(\varpi,\theta)$, $\varsigma_n(\varpi)$ and $\zeta_n(\theta)$, respectively, for the rest of this report.

D. Leakage power modeling

The analytical models of two major leakage currents, gate tunneling and subthreshold leakage currents, will be introduced in this section. The leakage current of each functional gate is input pattern dependent [11]. By applying different input patterns to different types of gates, their average leakage currents are measured to construct a set of cell leakage powers by using HSPICE. Then, the fitting model of empirical current for each type of gates is built by using the minimal least square fitting. The maximum error of fitting models compared with the results of HSPICE is less than 2%.

(1) Gate tunneling leakage current

Since the variations of gate leakage current are excessively sensitive to the variations of oxide thickness, the influence of channel length variations can be securely ignored [10]. Thus, the gate tunneling leakage current for a specific type of gate can be model as [11]

$$I_g=a_0\exp\left(a_1T_{ox}\right), \tag{10}$$

where a_0 and a_1 are fitting constants.

By substituting equations (9) into equation (10) and multiplying it by the supply voltage V_{dd}, the stochastic gate tunneling leakage power for a specific type of gate located at $\left(x^*,y^*\right)$ can be expressed as

$$P_g(x^*,y^*,\varsigma)=\overline{P}_g\exp(\overline{a}_1\varsigma^{T\mathrm{f}^*}), \qquad (11)$$

where $\overline{P}_g=\overline{a}_0V_{dd}$, a_0 and a_1 are known values, $\varsigma=\left[\varsigma_1,\varsigma_2,\cdots,\varsigma_{N_{T_{ox}}}\right]^T$, $\mathrm{f}^*=\left[f_1^*,\cdots,f_n^*\cdots,f_{N_{T_{ox}}}^*\right]^T$, and each $f_n^*=\sqrt{\beta_n}f_n\left(x^*,y^*\right)$.

(2) Sub-threshold leakage current

The sub-threshold leakage current is temperature dependent. For simplicity, we apply the following empirical form introduced in [26] at a suitable reference temperature.

$$I_s=b_0\exp(b_1L_{ch}+b_2L_{ch}^2), \qquad (12)$$

where b_0, b_1 and b_2 are fitting constants.

Substituting equations (8) into equation (12) and multiplying it by V_{dd} , the sub-threshold leakage power for a specified type of gate located at $\left(x^*,y^*\right)$ can be given as

$$P_s(x^*,y^*,\zeta)=\overline{P}_s\exp(\overline{b}_1\zeta^T\mathbf{q}^*+\overline{b}_2\zeta^T\mathbf{A}^*\zeta), \qquad (13)$$

where $\overline{P}_s=\overline{b_0}V_{dd}$, b_0, b_1 and b_2 are known values, $\zeta=\left[\zeta_1,\zeta_2,\cdots,\zeta_{N_{L_{ch}}}\right]^T$, $\mathrm{q}^*=\left[q_1^*,\cdots,q_m^*,\cdots,q_{N_{L_{ch}}}^*\right]^T$, A^* is a $N_{L_{ch}}\times N_{L_{ch}}$ symmetric matrix with each entry $A_{nl}^*=2^{\delta_{nl}-1}q_n^*q_l^*$, and each $q_m^*=\sqrt{\chi_m}q_m\left(x^*,y^*\right)$.

E. Stochastic Galerkin procedure via Hermite polynomial chaos

The Taylor expansion method has been widely used in the statistical timing and circuit performance analysis [16], [17]. However, the assumption of small variation of the desired solution with respect to the random variables is not appropriate for approximating the temperature distribution, because the leakage power exponentially depends on the physical parameters and the temperature is directly affected by the leakage power. With a similar situation, the inaccuracy of the second order Taylor expansion method for solving a power grid system with variations of physical parameters for log-normal leakage currents was indicated by [18].

On the contrary, the PC based method [12] is adopted because it can handle the desired solution with large variation with respect to the random variables and can achieve a minimal mean square error approximation. Moreover, the projected deterministic heat transfer equations in this work are un-coupled for PCs. Hence, the efficiency is equal to applying the Taylor expansion method to the stochastic heat transfer equations (3)–(4).

(1) Stochastic Galerkin procedure

With $\{\xi_n\}_{n=1}^{N_{KL}}$, a set of N_{KL}-dimensional Hermite Polynomial Chaoses (H-PCs) [12] can be constructed to serve as bases to expand a general second-order random process $u(\varpi,\theta)$. According to the theorem of Cameron and Martin [25], the random process of rising temperature distribution $T(\mathbf{r},\varpi,\theta)$ can be approximated as

$$T(\mathbf{r},\varpi,\theta) \approx \sum_{k=0}^{N_{PC}} T_k(\mathbf{r})\Phi_k(\xi), \qquad (14)$$

where each $T_k(\mathbf{r})$ is the projected temperature coefficient function of the k-th H-PC, $\Phi_k(\xi)$ is the k-th H-PC4, and N_{PC} is the truncation number. The relation between N_{PC} and $N_{L_{ch}}$ and $N_{T_{ox}}$ will be described in section 四、E.(4).

The stochastic Galerkin projection is executed as follows. Due to the limitation of space, the detail derivation is ignored.

1) Obtain the residual functions by substituting equation (14) into equations (3) and (4).

2) Enforce residual functions to be orthogonal to each H-PC.

Then, we obtain the following decoupled deterministic heat transfer equations for solving each $T_k(\mathbf{r})$ for each different k.

$$\kappa\nabla^2 T_k(\mathbf{r}) = -\frac{p_k(\mathbf{r})}{R\{\Phi_k^2(\xi)\}}, \qquad (15)$$

subject to the boundary condition

$$\kappa\frac{\partial T_k(\mathbf{r}_{bs})}{\partial n_{bs}} + h_{bs}T_k(\mathbf{r}_{bs}) = \hat{f}_{bs}(\mathbf{r}_{bs})\delta_{0k} \text{ for each } bs, \qquad (16)$$

where $p_k(\mathbf{r}) = E\{p(\mathbf{r},\xi)\Phi_k(\xi)\}$ is equal to

$$p_k(\mathbf{r}) = p_d(\mathbf{r},t)\delta_{0k} + p_{gk}(\mathbf{r}) + p_{sk}(\mathbf{r}). \qquad (17)$$

Here, $p_{gk}(\mathbf{r}) = E\{p_g(\mathbf{r},\varsigma)\Phi_k(\xi)\}$ and $p_{sk}(\mathbf{r}) = E\{p_s(\mathbf{r},\zeta)\Phi_k(\xi)\}$ are the projected gate-leakage and subthreshold-leakage power density profiles of the k-th H-PC, respectively. The term δ_{0k} in both equations (16) and (17) is because $E\{\Phi_k(\xi)\} = \delta_{0k}$ [12]. After $p_k(\mathbf{r})$ is calculated, any existing deterministic thermal simulator [2]–[8] can be utilized to obtain each $T_k(\mathbf{r})$.

The above un-coupled deterministic heat transfer equations have an advantage for both numerical and analytical thermal simulators. For example, if the numerical simulators [2]–[6] are employed to solve the deterministic heat transfer equations, the system matrices of the above deterministic heat transfer equations are the same. Hence, the system matrices handling, such as the LU decomposition [6], building the multi-grid cycle [4], and setting up the tri-diagonal matrix in each direction [2], can be performed only once. After that, all deterministic heat transfer equations can share the same post-process matrix to obtain the solution. In this work, we utilize an efficient early-stage thermal simulator [8] to serve as the deterministic thermal simulator.

The mean and variance profiles of the steady-state temperature can be obtained as

$$E\{T(\mathbf{r},\varpi,\theta)+T_a\} \approx T_0(\mathbf{r})+T_a, \qquad (18)$$

$$Var\{T(\mathbf{r},\varpi,\theta)+T_a\} \approx \sum_{k=1}^{N_{PC}} T_k^2(\mathbf{r})E\left\{\Phi_k^2(\xi)\right\}. \qquad (19)$$

Note that only one deterministic heat transfer equation is needed to solve for obtaining the spatial mean temperature distribution.

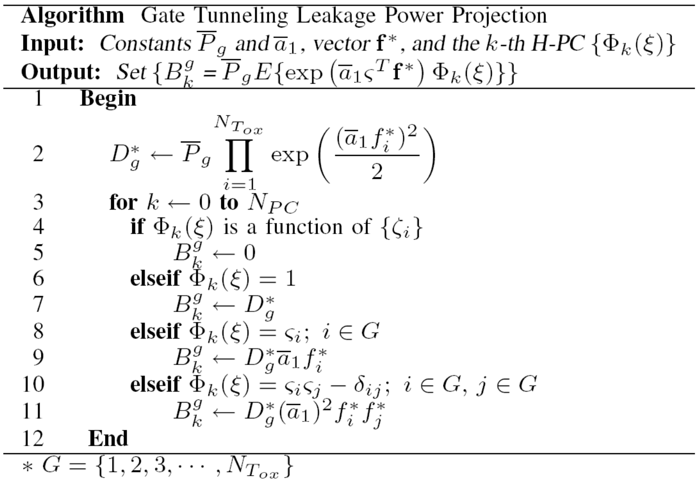
Two algorithms are proposed in the following two subsections to calculate the projected leakage powers for a specific type of gate located at arbitrary position of the die up to the second order of H-PCs. By those two algorithms, $p_k(\mathbf{r})$ can be obtained for solving $T_k(\mathbf{r})$.

(2) Gate leakage power projection

By using equation (11), the projected gate tunneling leakage power of the k-th H-PC for a specific type of gate located at a reference position $\left(x^*, y^*\right)$ is

$$E\left\{P_g(x^*, y^*, \varsigma)\Phi_k(\xi)\right\} = \overline{P}_g E\left\{\exp(\overline{a}_1\varsigma^T\mathbf{f}^*)\Phi_k(\xi)\right\}. \qquad (20)$$

Fig. 4 shows an algorithm for calculating equation (20) up to second order of H-PCs. Steps $4 \sim 5$ are owing to the independence of $\left\{\varsigma_i\right\}$ and $\left\{\zeta_i\right\}$. The rest steps of Fig. 4 can be derived by utilizing 0-th, 1-th and 2-th derivatives of the moment generating function of independent standard normal random variables.



```
Algorithm  Gate Tunneling Leakage Power Projection
Input:  Constants $\overline{P}_g$ and $\overline{a}_1$, vector $\mathbf{f}^*$, and the $k$-th H-PC $\{\Phi_k(\xi)\}$
Output:  Set $\{B_k^g = \overline{P}_g E\{\exp(\overline{a}_1\varsigma^T\mathbf{f}^*)\,\Phi_k(\xi)\}\}$
 1   Begin
 2       $D_g^* \leftarrow \overline{P}_g \prod_{i=1}^{N_{T_{ox}}} \exp\left(\frac{(\overline{a}_1 f_i^*)^2}{2}\right)$
 3       for $k \leftarrow 0$ to $N_{PC}$
 4         if $\Phi_k(\xi)$ is a function of $\{\zeta_i\}$
 5             $B_k^g \leftarrow 0$
 6         elseif $\Phi_k(\xi) = 1$
 7             $B_k^g \leftarrow D_g^*$
 8         elseif $\Phi_k(\xi) = \varsigma_i;\ i \in G$
 9             $B_k^g \leftarrow D_g^*\overline{a}_1 f_i^*$
10         elseif $\Phi_k(\xi) = \varsigma_i\varsigma_j - \delta_{ij};\ i \in G,\ j \in G$
11             $B_k^g \leftarrow D_g^*(\overline{a}_1)^2 f_i^* f_j^*$
12    End
* $G = \{1, 2, 3, \cdots, N_{T_{ox}}\}$
```

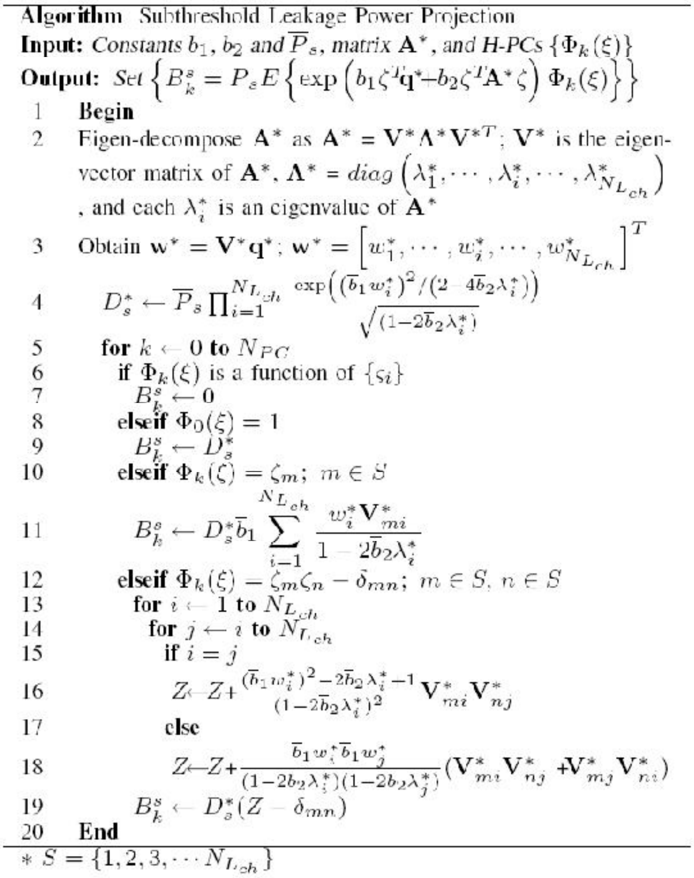
Fig. 4. Gate tunneling leakage power projection algorithm

(3) Sub-threshold leakage power projection

By using equation (13), the projected subthreshold leakage power of the k-th H-PC for a specific type of gate located at a reference position (x_; y_) is

$$E\left\{P_s\left(x^*, y^*, \zeta\right)\Phi_k\left(\xi\right)\right\} = \overline{P}_s E\left\{\exp\left(\overline{b}_1\zeta^T \mathrm{q}^* + \overline{b}_2\zeta^T \mathrm{A}^*\zeta\right)\Phi_k\left(\xi\right)\right\} \qquad (21)$$

Fig. 5 shows an algorithm for calculating equation (21) up to the second order of H-PCs. Due to the limitation of space, the derivation is ignored.



```
Algorithm Subthreshold Leakage Power Projection
Input: Constants $b_1$, $b_2$ and $\overline{P}_s$, matrix $\mathbf{A}^*$, and H-PCs $\{\Phi_k(\xi)\}$
Output: Set $\left\{B_k^s = P_s E\left\{\exp\left(b_1\zeta^T\mathbf{q}^* - b_2\zeta^T\mathbf{A}^*\zeta\right)\Phi_k(\xi)\right\}\right\}$
Begin
Eigen-decompose $\mathbf{A}^*$ as $\mathbf{A}^* = \mathbf{V}^*\mathbf{\Lambda}^*\mathbf{V}^{*T}$; $\mathbf{V}^*$ is the eigen-vector matrix of $\mathbf{A}^*$, $\mathbf{\Lambda}^* = diag\left(\lambda_1^*, \cdots, \lambda_i^*, \cdots, \lambda_{N_{L_{ch}}}^*\right)$, and each $\lambda_i^*$ is an eigenvalue of $\mathbf{A}^*$
Obtain $\mathbf{w}^* = \mathbf{V}^*\mathbf{q}^*$; $\mathbf{w}^* = \left[w_1^*, \cdots, w_i^*, \cdots, w_{N_{L_{ch}}}^*\right]^T$
  $D_s^* \leftarrow \overline{P}_s \prod_{i=1}^{N_{L_{ch}}} \frac{\exp\left((\overline{b}_1 w_i^*)^2/(2-4\overline{b}_2\lambda_i^*)\right)}{\sqrt{(1-2\overline{b}_2\lambda_i^*)}}$
  for $k \leftarrow 0$ to $N_{PC}$
    if $\Phi_k(\xi)$ is a function of $\{\varsigma_i\}$
      $B_k^s \leftarrow 0$
    elseif $\Phi_0(\xi) = 1$
      $B_k^s \leftarrow D_s^*$
    elseif $\Phi_k(\zeta) = \zeta_m$;  $m \in S$
      $B_k^s \leftarrow D_s^*\overline{b}_1 \sum_{i=1}^{N_{L_{ch}}} \frac{w_i^*\mathbf{V}_{mi}^*}{1-2\overline{b}_2\lambda_i^*}$
    elseif $\Phi_k(\xi) = \zeta_m\zeta_n - \delta_{mn}$;  $m \in S$, $n \in S$
      for $i \leftarrow 1$ to $N_{L_{ch}}$
        for $j \leftarrow i$ to $N_{L_{ch}}$
          if $i = j$
            $Z \leftarrow Z + \frac{(\overline{b}_1 w_i^*)^2 - 2\overline{b}_2\lambda_i^* - 1}{(1-2\overline{b}_2\lambda_i^*)^2}\mathbf{V}_{mi}^*\mathbf{V}_{nj}^*$
          else
            $Z \leftarrow Z + \frac{\overline{b}_1 w_i^* \overline{b}_1 w_j^*}{(1-2b_2\lambda_i^*)(1-2b_2\lambda_j^*)}(\mathbf{V}_{mi}^*\mathbf{V}_{nj}^* + \mathbf{V}_{mj}^*\mathbf{V}_{ni}^*)$
      $B_k^s \leftarrow D_s^*(Z - \delta_{mn})$
End
* $S = \{1, 2, 3, \cdots N_{L_{ch}}\}$
```

Fig. 5. Sub-threshold leakage power projection algorithm.

As indicated in [11], the number of reference points for modeling physical parameters can be much less than the number of gates while maintaining an acceptable accuracy. The simulated chip is divided into Ng grids for modeling physical parameters, and the central point of each grid is set to be a reference point $\left(x^*, y^*\right)$. Gates located in the same grid share the same modeled physical parameters; hence, they have the same A^*. Therefore, the number of eigen-decompositions for all A^* is N_g instead of the number of gates. Moreover, the eigenfunctions and eigenvalues of the channel length only depend on the spatial covariance function of the channel length, and each A^* can be known after the information of spatial covariance function is given. Therefore, the eigen-decomposition of each A^* and the projected power for each type of gate can be calculated before the thermal simulation. After the projected power for each type of gate at each grid is obtained, the rest computational cost for obtaining the steady-state sub-threshold-leakage power density profile of the k-th H-PC is O(#Gates).

(4) Truncated number of H-PCs

The original truncated number of H-PCs is [12]

$$N_{pc} = 1 + \sum_{n=1}^{p} \frac{1}{n!} \prod_{r=0}^{n-1} \left(N_{KL} + r\right), \qquad (22)$$

where p is the order of H-PCs, and $N_{KL} = N_{T_{ox}} + N_{L_{ch}}$.

The projection values of $P_g\left(x,y,\varsigma\right)$ and $P_s\left(x,y,\zeta\right)$ upon the H-PC which simultaneously contains random variables in $\{\varsigma_i\}$ and $\{\zeta_i\}$ are equal to zeros. Therefore, the number of H-PCs is reduced to

$$N_{pc}=1+\sum_{n=1}^{p}\frac{1}{n!}\prod_{r=0}^{n-1}\left(N_{T_{ox}}+r\right)+\sum_{n=1}^{p}\frac{1}{n!}\prod_{r=0}^{n-1}\left(N_{L_{ch}}+r\right). \tag{23}$$

Since reduced truncated number is much less than the original truncated number, the computational effort is significantly reduced.

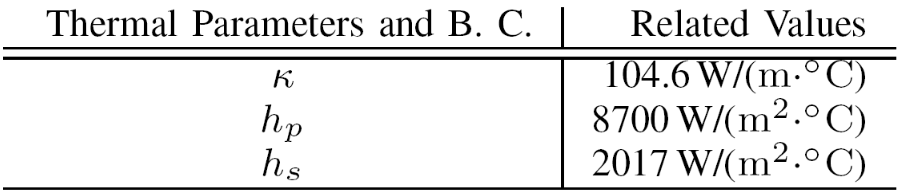
F. Experimental Results

Our stochastic thermal simulator is implemented in C++ language and tested on a HPxw9300 workstation with 16GB memory. The die size is 5mm × 5mm × 0.5 mm. The device junction depth is set to be 20nm which is the nominal value of the device junction depth for the 65nm technology [9]. The test chip floorplan is shown as rectangular blocks in Fig. 7(b). Numerous functional gates with the 65 nm technology are inserted into each rectangular block of Fig. 7(b), and the number of functional gates on the test chip is around 4.7 millions. The internal gates of each rectangular block in Fig. 7(b) are not shown for the sake of clarity.

The nominal value of oxide thickness is 1.4 nm and the 3σ values of parameter variations for the channel length and the oxide thickness are 20% of their nominal values. Both η_x / L_x and η_y / L_y are set to 0.31 which means the correlation between two devices located half of the chip dimension away in either the x-direction or the y-direction is 0.2 [20]. The number of reference points is set to be 16 for the parameter modeling of the channel length and the oxide thickness. Based on the criterion stated in section IV-B, the truncation points of KL expansions for the channel length and the oxide thickness are chosen as $N_{L_{ch}}$ = 82 and $N_{T_{ox}}$ = 82, respectively.

The values of hp and hs for executing [8] are obtained as follows. Based on the same setting of chip geometry as [7], hp is obtained as $8700W/\left(m^2\cdot{}^{\circ}C\right)$. The value of h_s is got by using the modeling techniques of the equivalent thermal resistance for the C4/CBGA package [27] and effective thermal conductivity for interconnect layers [7].

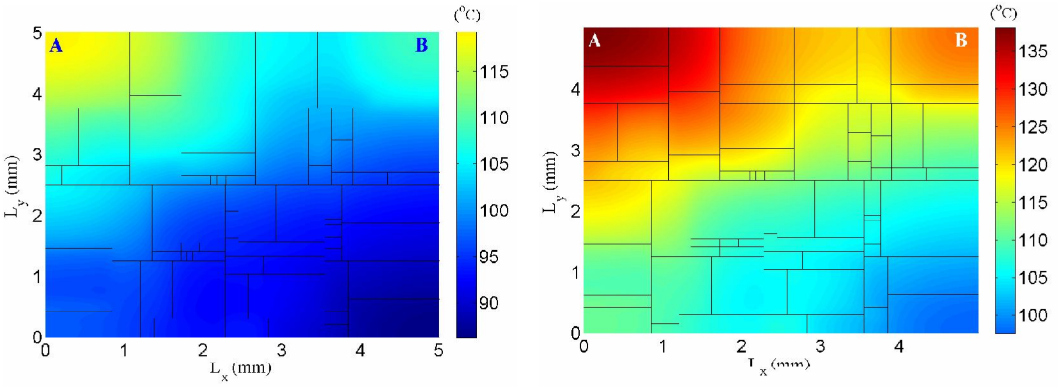
To set the thermal conductivity of die at the roughly steady-state average mean temperature of die, we apply the 1-D thermal model shown in Fig. 2 and the following iteratively computation scheme. Initially, T_{avg} is set to be T_a, then the initial value of R_{die} can be obtained. With this R_{die} and calculated P_T which can be got by using 0-th order projected powers of H-PCs sated in section VI-B and VI-C, we update T_{avg} by the 1-D thermal model. The above procedure is repeated until T_{avg} converges. Here, the room temperature T_a is set to be $27^{\circ}C$. With the above calculation, the related thermal parameters and boundary conditions for executing the deterministic simulator [8] are summarized in Fig. 6. The boundary condition of each vertical surface is set to be isothermal [7], [8]. The top surface of the simulated die is divided into 1024×1024 grids for solving the deterministic heat transfer equations with respect to each H-PC.



Thermal Parameters and B. C.	Related Values
κ	104.6 W/(m·°C)
h_p	8700 W/(m^2·°C)
h_s	2017 W/(m^2·°C)

Fig. 6. Thermal conductivity of the die calculated at $T_{avg} + T_a$, and the equivalent heat transfer coefficients hp and hs for the primary and the secondary heat flow paths, respectively.

(1) Accuracy and efficiency



(a) **(b)**

Fig. 7. The temperature distribution at the top surface of the test die. (a) The nominal temperature distribution and (b) the spatial mean temperature distribution.

The Monte Carlo (MC) method with 105 samples and 256 reference points for modeling parameters are used as the reference solution to demonstrate the accuracy. As shown in Table I, the maximum errors of our simulator are less than 2% in both the spatial mean and the spatial standard deviation distributions with NKL being equal to 164 and the order of H-PCs being equal to 1. The runtime is only 113 seconds. Since Ng is set to be 16 in our simulator, the result demonstrates that Ng can be quite small without sacrificing the accuracy.

The average mean temperature at the top surface of the die calculated by our simulator is $114.73^{\circ}C$. Note that, the average mean temperature at the top surface of the die calculated by the 1-D thermal model is $114.75^{\circ}C$, which is consistent with the value got by our simulator. This verifies the ability of the 1-D thermal model for predicting the average steady-state mean temperature of the die. Thus, the thermal parameters are set at an accurate average steady-state mean temperature of the die.

To demonstrate the efficiency of the proposed method, the runtime comparison between the proposed method and the Monte Carlo method is shown in Table I. Here, the number of sampling times for the Monte Carlo method is set for achieving the same standard deviation error level as our simulator. The results show that our simulator can be orders of magnitude faster than the Monte Carlo method.

It can be observed that the maximum error of the mean profile of the temperature only depends on N_{KL} rather than the order of H-PCs. However, the maximum error of the spatial standard profile of the temperature relies not only on N_{KL} but also on the order of H-PCs. As shown in Table I, using the first order of H-PCs with large N_{KL} can provide an accurate solution and the complexity is linear to N_{KL}. Using the second order of H-PCs with small N_{KL} can also provide an accurate solution but the number of the N_{PC} increases quadratically.

Based on the above observation, the following strategy can be used to further improve the accuracy without sacrificing the efficiency. After the initial N_{KL} is decided by the criterion stated in section IV-B, the temperature coefficient function for each first order polynomial chaos can be obtained. Then, the temperature coefficient function of the second order polynomial chaos with respect to the decreasing order of eigenvalues one by one is obtained until there is no significant change of the standard deviation profile of the temperature.

(2) Deterministic v.s. stochastic thermal simulators

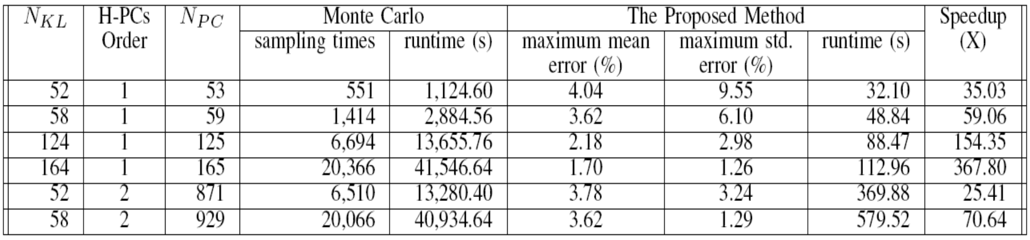
The nominal and mean temperature profiles on the top surface of the die are shown in Fig. 7(a) and (b), respectively. The difference between them is over 16%. This indicates that the deterministic thermal simulator with the nominal power underestimates the hottest value and profile of the die temperature.

(3) Without or with including the effect of spatial correlation

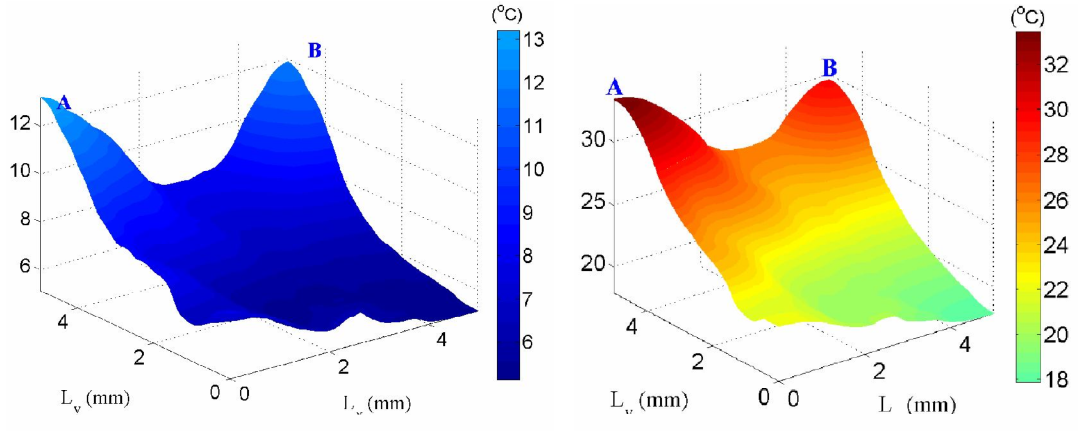
Fig. 8 reveals the dramatic difference of the standard deviation profiles for the temperature between the result considering the spatial correlation of physical parameters and the result ignoring the spatial correlation of physical parameters. Although their spatial mean temperature distributions are equal because of equation (18), their spatial standard deviations profile of temperature are drastically different. The values presented in Fig. 8(b) are 3~4 times of Fig. 8(a). Hence, the spatial correlation of the physical parameter should be taken into account in the stochastic thermal analysis.

TABLE I

ACCURACY AND EFFICIENCY COMPARED WITH THE MONTE CARLO METHOD.



N_{KL}	H-PCs Order	N_{PC}	Monte Carlo		The Proposed Method			Speedup (X)
			sampling times	runtime (s)	maximum mean error (%)	maximum std. error (%)	runtime (s)	
52	1	53	551	1,124.60	4.04	9.55	32.10	35.03
58	1	59	1,414	2,884.56	3.62	6.10	48.84	59.06
124	1	125	6,694	13,655.76	2.18	2.98	88.47	154.35
164	1	165	20,366	41,546.64	1.70	1.26	112.96	367.80
52	2	871	6,510	13,280.40	3.78	3.24	369.88	25.41
58	2	929	20,066	40,934.64	3.62	1.29	579.52	70.64

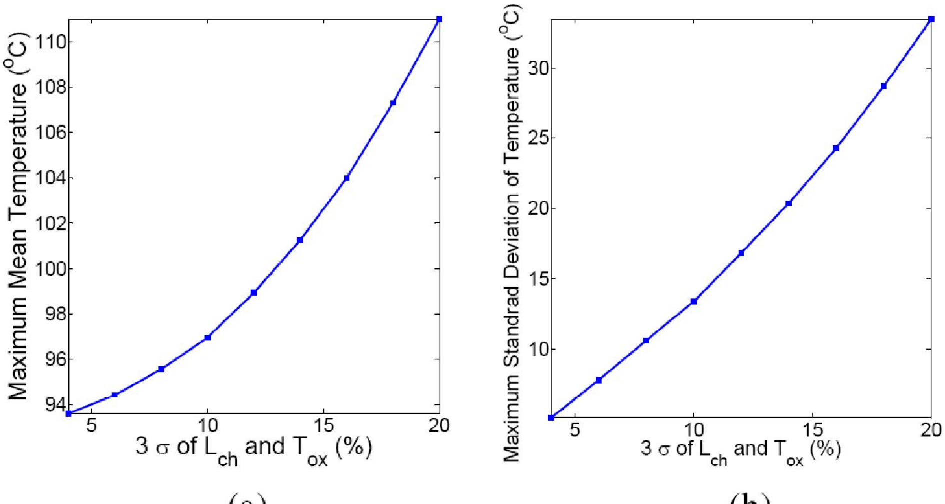


(a) **(b)**

Fig. 8. The spatial standard deviations for the temperature distribution at the top surface of the test die. (a) The spatial standard deviations without including the effect of spatial correlation and (b) the spatial standard deviations with including the effect of spatial correlation.

(4) Temperature variation trend with respect to variation of physical parameters

To further study the trend of temperature variation with respect to the variation of physical parameters, we sweep the 3σ ranges of channel length and oxide thickness and show the corresponding maximum mean and maximum standard deviation of die temperature in Fig. 9. As we can see, both of the maximum mean and the maximum standard deviation are exponentially dependent on the 3σ ranges of channel length and oxide thickness.



(a) (b)

Fig. 9. (a) The maximum mean of die temperature. (b) The maximum standard deviation of die temperature.

五、結論與討論

In this report, we have developed a stochastic thermal simulator with considering spatial correlated within-die process variations. The experimental results have demonstrated that our simulator can efficiently provide very accurate estimates. Our simulator can readily provide a simulating kernel of the elector-thermal simulating loop. Our future work is to combine our simulator with the elector-thermal simulating loop for providing a more accurate thermal estimation under the spatial correlated within die process variations.

六、成果

In the last year of this project, we have published three international conference papers [R1, R2, R6] and two domestic conference papers [R4, R5], one regular paper [R3] has been accepted by TVLSI, and two international conference papers [R7, R8] has been accepted by ASPDAC 2009.

[R1] Pei-Yu Huang, Chih-Kang Lin, and Yu-Min Lee, "Full-Chip Thermal Analysis via Generalized Integral Transforms", *the 14th Workshop on Synthesis and System Integration of Mixed Information Technologies (SASIMI)*, 2007.

[R2] Pei-Yu Huang, Chih-Kang Lin, and Yu-Min Lee, "Full-Chip Thermal Analysis for the Early Design Stage via Generalized Integral Transforms", *the 13th Asia and South Pacific Design Automation Conference (ASPDAC) 2008*, pp. 462-7, 2008.

[R3] Pei-Yu Huang and Yu-Min Lee, "Full-Chip Thermal Analysis for the Early Design Stage via Generalized Integral Transforms", accepted by *IEEE Transactions on Very Large Scale Integration Systems (TVLSI)*.

[R4] Pei-Yu Huang, Jia-Hong Wu, Yu-min Lee, and Huai-Chung Chang, "Stochastic Thermal Simulation Considering With-in Die Process Variation", *the 19th VLSI Design/ CAD Symposium (VLSI/CAD 2008).*

[R5] Shih-An Yu, Pei-Yu Huang and Yu-Min Lee, "Power Optimization in 3D ICs Considering Process Variations and Thermal Effect", *the 19th VLSI Design/ CAD Symposium (VLSI/CAD 2008).*

[R6] Pei-Yu Huang, Chih-Kang Lin, and Yu-Min Lee, "Hierarchical Power Delivery Network Analysis using Markov Chains", *IEEE International SOC Conference* (*SOCC*) *2007.*

[R7] Pei-Yu Huang, Jia-Hong Wu and Yu-Min Lee, "Stochastic Thermal Simulation Considering Spatial Correlated Within-Die Process Variations", *to appear in Asia South Pacific Design Automation Conference (ASPDAC) 2009.*

[R8] Shih-An Yu, Pei-Yu Huang and Yu-Min Lee, "A Multiple Supply Voltage Based Power Reduction Method In 3-D ICs Considering Process Variations And Thermal Effects", *to appear in Asia South Pacific Design Automation Conference (ASPDAC) 2009*.

七、參考文獻

[1] S. Borkar, T. Karnik, S. Narendra, J. Tschanz, A. Keshavarzi, and V. De, “Parameter variations and impact on circuits and microarchitecture,” *in Proc. Des. Autom. Conf.*, 2003, pp. 338–42.

[2] T. Y. Wang and C. C. P. Chen, “Thermal-ADI: a linear-time chip-level thermal simulation algorithm based on alternating-direction implicit (ADI) method,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 11, no. 4, pp. 691–700, Aug. 2003.

[3] T. Y. Wang and C. C. P. Chen, “SPICE-compatible thermal simulation with lumped circuit modeling for thermal reliability analysis based on model reduction,” *in Proc. Int. Symp. Quality Electron. Des.*, 2004, pp. 357–62.

[4] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, “IC thermal simulation and modeling via efficient multigrid-based approaches,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 9, pp. 319–26, Sep. 2006.

[5] Y. Yang, Z. Gu, C. Zhu, R. P. Dick, and Li Shang, “ISAC: Integrated space and-time-adaptive chip-package thermal analysis,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 1, pp. 86–99, Jan. 2007.

[6] 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. 501–13, May 2006.

[7] Y. Zhan and S. S. Sapatnekar, “High efficiency Green function-based thermal simulation algorithms,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 26, no. 9, pp. 1661–75, Sep. 2007.

[8] P. Y. Huang, C. K. Lin, and Y. M. Lee, “Full-chip thermal analysis for the early design stage via generalized integral transforms,” *in Proc. Asia and South Pacific Des. Autom. Conf.*, 2008, pp. 462–7.

[9] F. Lallement, B. Duriee, A. Grouillet, F. Amaud, B. Tavel, F. Wacquant, P. Stalk, M. Woo, Y. Erokhin, J. Scheuer, L. Gadet, J. Weeman, D. Distaso, D. Lenoble, “Ultra-low cost and high performance 65nm CMOS device fabricated with plasma doping,” *in Symp. VLSl Technol. Dig. Tech. Papers*, 2004, pp. 178–9.

[10] A. Srivastava, D. Sylvester, and D. Blaauw, Statistical Analysis and Optimization for VLSI: Timing and Power, Springer-Verlag, 2004.

[11] H. Chang and S. S. Sapatnekar, “Prediction of leakage power under process uncertainties,” *ACM Trans. Design Autom. Electron. Syst.*, vol. 12, no. 2, article 12, Apr. 2007.

[12] R. G. Ghanem and P. D. Spanos, Stochastic Finite Elements: A Spectral Approach, revised edition, Springer-Verlag, 2003.

[13] J.-L. Tsai, C. C.-P. Chen, G. Chen, B. Goplen, H. Qian, Y. Zhan, S.-M. Kang, M. D. F. Wong and S. S. Sapatnekar, “Temperature-aware placement for SOCs,” *Proc. IEEE*, vol. 94, no. 8, pp. 1502–18, Aug. 2006.

[14] G. A. F. Seber, Multivariate Observations, John Wiley & Sons, Inc., 2004.

[15] S. Bhardwaj, S. Vrudhula, P. Ghanta, and Y. Cao, “Modeling of intradie process

variations for accurate analysis and optimization of nanoscale circuits," *in Proc. Des. Autom. Conf.*, 2006, pp. 791–6.

[16] X. Ye, P. Li, and F. Liu, "Practical Variation-Aware Interconnect Delay and Slew Analysis for Statistical Timing Verification," *Proc. Int. Conf. on Comput. -Aided Des.*, pp. 54-59, Nov. 2006.

[17] H. Chang and S. Sapatankar, "Statistical timing analysis under spatial correction," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 24, no. 9, pp. 1467-1482, Sept. 2005.

[18] N. Mi, J. Fan, S. X.-D. Tan, Y. Cai, and X. Hong, "Statistical Analysis of On-Chip Power Delivery Networks Considering Lognormal Leakage Current Variations with Spatial Correlation," *IEEE Trans. on Circuits and Syst.*, accepted for future publication.

[19] M. Loeve, Probability Theory, D. Van Nostrand Company Inc., 1960.

[20] B. Cline, K. Chopra, D. Blaauw, and Y. Cao, "Analysis and modeling of CD variation for statistical static timing," *in Proc. Int. Conf. on Comput.-Aided Des.*, 2006, pp. 60–66.

[21] F. Liu, "A general framework for spatial correlation modeling in VLSI design," *in Proc. Des. Autom. Conf.*, 2007, pp. 817–22.

[22] J. Xiong, V. Zolotov, and L. He, "Robust extraction of spatial correlation," *in Int. Symp. Phys. Des.*, 2005, pp. 2–9.

[23] D. Zhang and Z. Lu,"An efficient, high-order perturbation approach for flow in random porous media via Karhunen-Lo_eve and polynomial expansions," *J. Comput. Phys.*, vol. 149, no. 2, pp. 773–94, Mar. 2004.

[24] C. Schwab and R. A. Todor, "Karhunen-Loeve approximation of random fields by generalized fast multipole methods," *J. of Comput. Phys.*, vol. 217, issue 1, pp. 100–22, Sep. 2006.

[25] R. H. Cameron and W. T. Martin, "The orthogonal development of nonlinear functionals in series of Fourier-Hermite functionals," *Ann. of Math.*, vol. 48, no. 2, pp. 385–92, Apr. 1947.

[26] R. Rao, A. Srivastava, D. Blaauw, and D. Sylvester, "Statistical analysis of subthreshold leakage current for VLSI circuits," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 12, no. 2, pp. 131–9, Feb. 2004.

[27] C. Lasance, H. Vinke, H. Rosten, and K.-L. Weiner, "A novel approach for the thermal characterization of electronic parts," *in IEEE Semi-Therm Symp.*, 1995, pp. 1–9.

Full-Chip Thermal Analysis for the Early Design Stage via Generalized Integral Transforms

Pei-Yu Huang, Chih-Kang Lin, and Yu-Min Lee, *Member*, *IEEE*
Department of Communication Engineering, National Chiao Tung University Hsinchu 300, Taiwan
pey.cm93g@nctu.edu.tw, is84013@cis.nctu.edu.tw and yumin@cm.nctu.edu.tw

Abstract— **The capability of predicting the temperature profile is critically important for circuit timing estimation, leakage reduction, power estimation, hotspot avoidance, and reliability concerns during modern IC designs.**

This paper presents an accurate and fast analytical full-chip thermal simulator for the early-stage temperature-aware chip design. By using the technique of generalized integral transforms (GIT), our proposed method can accurately estimate the temperature distribution of full-chip with very small truncation points of bases in the spatial domain. We also develop a fast Fourier transform (FFT) like evaluating algorithm to efficiently evaluate the temperature distribution. Experimental results confirm that our GIT based analyzer can achieve an order of magnitude speedup compared with a highly efficient Green's function based method.

I. Introduction

The power density of VLSI circuits increases monotonously as the CMOS technology continuously scales down. Because the power dissipated by circuits converts into heat, as a result, it raises the temperature of dies and induces hot spots. These thermal-related phenomena significantly degrade the performance and reliability of circuits [1]–[6], To precisely predict thermal impacts on design performance, an efficient and accurate thermal analyzer is necessary in the temperature-aware design flow because it is usually a part of simulation kernel in the optimization loop and need to be executed numerous times.

Essentially, existing thermal simulators can be categorized into two classes, numerical and analytical methods. The numerical methods apply the finite difference method (FDM) or finite element method (FEM) to transfer heat equations to RC network equations. Based on the RC network equations, several methods are proposed to improve the run time. For example, the alternating-direction-implicit based method [1], the model order reduction based method [2], and the multi-grid method [3]. Because of the flexibility for dealing with complicated structure, the numerical framework is suitable for the back-end stage of design flow such as the post layout thermal verification.

As pointed out in [4], the temperature-aware design should be brought to the early design stage such as thermal-aware floorplanning and placement. To give a reasonably accurate temperature prediction with little computational effort, they proposed an accurate compact thermal model for modeling equivalent heat transfer coefficients of the pre-layout package and interconnect layers for the boundary conditions of die, and provided a numerical method for the temperature calculating of die.

Although numerical methods can be directly applied to simulate the temperature distribution of the model proposed in [4], they are not suitable for the early temperature-aware design stage because they require the volume meshing of entire substrate even if the devices are usually built within a thin layer close to the top surface of die, and the material of substrate can be treated as homogeneous during the early design stage [5]. Because of the volume meshing, a huge set of linear equations for the uninterested temperature in substrate still need to be handled even if only the temperature distribution close to the device layer is of interest.

On the contrary, analytical methods are good candidates for the early design stage because they avoid directly performing the volume meshing of entire substrate and have closed-form representations for the temperature distribution. One analytical category of thermal solvers is the Green's function based method [6]. In [6], the authors applied the Green's function to the time-independent Possion's equation and used fast Fourier transform (FFT) to evaluate the steady-state temperature distribution. Hence, the computational cost can be only $O(MN\log_2 MN)$, where M, and N are numbers of divisions in the power density map along x-, and y-directions, respectively. However, the convergent rate of their formulation is not fast enough because the generated cosine series based on time-independent Possion's equation [6] can not fully capture the transient characteristics of original heat equations. As shown in [6], the truncation points need to be large enough to achieve small relative error. Furthermore, it is only for steady state temperature calculation, but the transient analysis is also necessary while performing the dynamic thermal management and run-time thermal analysis [4]–[6].

To overcome these shortcomings, our major contributions are

- We improve the convergence rate of analytical solution for steady state temperature distribution and provide a transient temperature simulation by utilizing generalized integral transforms (GIT) [7] to construct a set of spatial bases and their corresponding time-varying coefficients. The proposed method can accurately estimate the temperature distribution of full-chip with very small truncation points (N_x and N_y) of bases in the spatial domain. The experimental results presented in section IV show that N_xN_y can be far less than MN without losing any accuracy compared with [6].
- We develop a FFT like evaluating algorithm to efficiently evaluate the temperature map of all grid cells, and its computational cost is in order of $O(MN\log_2 N_xN_y)$, where N_x, and N_y are truncation points of bases along x-, and y-directions, respectively.

The rest of this paper is organized as follows. First, the thermal modeling for the early design stage is introduced in section II. Then, the full-chip thermal simulation by using the GIT technique, and the proposed temperature evaluating algorithm are described in section III. Finally, the experimental results and conclusions are given in section IV and V, respectively.

II. Thermal modeling for the early design stage

The thermal model for the early design stage can be modeled as a compact structure which consists of the primary heat flow path, the secondary heat flow path, and the heat transfer characteristic of each macro/block on silicon die [4] as shown in Fig. 1. The primary heat flow path is composed of thermal interface material, heat spreader, and heat sink. The secondary heat flow path contains interconnect layers, I/O pads, and print circuit board

This work was supported in part by National Science Council of Taiwan, in part by MoE ATU Program, and in part by SoC Research Center of NCTU. We are grateful to the National Center for High-performance Computing for computer time and facilities.

978-1-4244-1922-7/08/$25.00 ©2008 IEEE

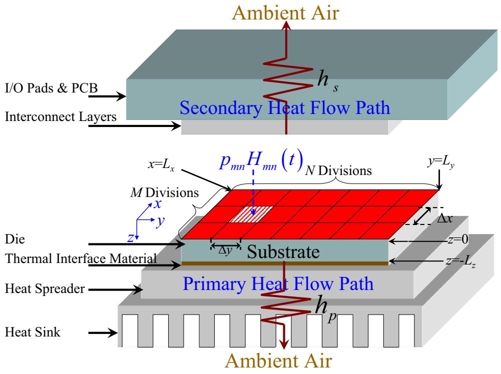


Fig. 1. Compact thermal model of the early design stage.

(PCB). The functional blocks on the die are modeled as many power sources attached on the top surface of die.

The rising temperature $T(\mathbf{r}, t)$ of die corresponding to the ambient temperature can be governed by the following heat transfer equations [5].

$$\nabla \cdot (\kappa(\mathbf{r})\nabla T(\mathbf{r}, t)) = \sigma(\mathbf{r})\frac{\partial T(\mathbf{r}, t)}{\partial t}; \mathbf{r} \in D \tag{1}$$

$$\kappa(\mathbf{r})\frac{\partial T(\mathbf{r}, t)}{\partial n_{b_s}} + h_{b_s}T(\mathbf{r}, t) = f_{b_s}(\mathbf{r}) \tag{2}$$

where $\mathbf{r} = (x, y, z)$, $\kappa(\mathbf{r})$ is the thermal conductivity (W/m·°C) of die, $\sigma(\mathbf{r})$ is the product of the density of matrial and specific heat (J/m^3·°C) of die, ∇ is the diverge operator, D=(0,L_x)×(0,L_y)×(−L_z,0) is the dimension of die, L_x and L_y are the lateral sizes of die, L_z is the thickness of die, h_{b_s} is the heat-transfer coefficient on the boundary surface, b_s, of die, $f_{b_s}(\mathbf{r})$ is the heat flux function on the boundary surface, and $\partial/\partial n_{b_s}$ is the differentiation along the outward direction normal to the boundary surface.

To provide reasonable accuracy with little computational effort during the early-stage temperature-aware optimization procedure, heat-transfer coefficients on the boundary surfaces of die should be appropriately modeled. Based on the model proposed in [4], the heat transfer coefficients of primary path can be equalized to an effective heat transfer coefficient h_p by combining the effect of each component on the primary path. Since the detail layout of interconnects is not available in the early design stage, [4] modeled the interconnect layer as an equivalent thermal resistance by estimating the density based on the regularity structure assumption of metal and dielectric material. With the model of each interconnect layer, the heat transfer coefficients of secondary path can be simplified to be an equivalent heat transfer coefficient h_s by stacking the thermal resistance of each layer with I/O pads and PCB.

Because of the chip and package structures, the area of vertical surface is strictly less than the area of horizontal surface and the thermal conductivity of air is much less than the values of primary and secondary heat transfer paths. Therefore, the boundary condition of vertical surface is set to be adiabatic [6]. The heat transfer characteristic of functional blocks on the die is modeled as an equivalent heat equation with many power generating sources attached on the top surface of die and substrate.

Finally, although in general, the thermal parameters, $\kappa(\mathbf{r})$ and $\sigma(\mathbf{r})$, of die are position-dependent, the variations of these thermal parameters are usually not significant, and as suggested in [4]–[6], these parameters can be treated as constants while performing the temperature-aware floor-planning and placement.

Based on the above model, the heat diffusion equations of die for the early design stage can be re-written as

$$\kappa\nabla^2 T(x, y, z, t) = \sigma\frac{\partial T(x, y, z, t)}{\partial t}; (x, y, z) \in D \tag{3}$$

$$\left.\frac{\partial T(x, y, z, t)}{\partial x}\right|_{x=0,L_x} = \left.\frac{\partial T(x, y, z, t)}{\partial y}\right|_{y=0,L_y} = 0 \tag{4}$$

$$\kappa\left.\frac{\partial T(x, y, z, t)}{\partial z}\right|_{z=-L_z} = h_p T(x, y, -L_z, t) \tag{5}$$

$$\kappa\left.\frac{\partial T(x, y, z, t)}{\partial z}\right|_{z=0} = h_s T(x, y, 0, t) + p(x, y, t) \tag{6}$$

Here, $p(x, y, t)$ is the power density (W/m^2) on the top surface of die and $T(x, y, z, 0)$ = 0.

By discretizing the power source plane on the top of die into MN grid cells, where M and N are numbers of divisons along x- and y-directions, respectively, the power density $p(x, y, t)$ can be rewritten as

$$p(x, y, t) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} p_{mn}\Pi_{mn}(x, y)H_{mn}(t), \tag{7}$$

where $\Pi_{mn}(x, y)$ is the indicate function with nonzero value equaling to 1 only when (x, y) is in $[m\Delta x, (m+1)\Delta x] \times [n\Delta y, (n+1)\Delta y]$, Δx=L_x/M, Δy=L_y/N, m and n are indices of divisions, and p_{mn} and $H_{mn}(t)$ are the average power density and the turning on/off function of grid cell (m, n), respectively.

As calculating the steady state temperature, $H_{mn}(t)$ is a unit step function. Otherwise, it is an instruction specified time interval function [3]. With above government equations, our goal is to get the rising temperature distribution of die corresponding to the ambient temperature.

III. Full-Chip Thermal Simulation

The computational procedure of GIT includes two steps [7]. In the beginning, a set of appropriate bases is generated by a system-compatible auxiliary problem. Several guidelines need to be followed for choosing this auxiliary problem. Firstly, the auxiliary problem should be as similar as the original problem. Secondly, the generated bases have to be completely ortho-normalized to ensure the convergence in mean of the approximated temperature distribution [7]. Finally, the ortho-normal bases should be time independent for efficiency consideration. After bases being constructed, the temperature distribution can be expressed by those bases with suitable time-varying coefficients.

In next two subsections, how to apply the above procedure to the full-chip thermal analysis will be described in detail. After that, the compact formula will be given to calculate the average steady-state temperature distribution, and its convergence rate improvement over the Green's function based method [6] will be pointed out. Finally, we will develop fast evaluating algorithms for our GIT based formulation, and utilize our method to perform the transient thermal simulation.

A. Auxiliary Problem for Generating Appropriate Spatial Bases

The auxiliary problem can be introduced by considering the homogeneous problem which the solution of temperature distribution satisfies homogeneous government equations (3)-(6) with $p(x, y, t)$ = 0. As stated in [7], the auxiliary problem can be the following Sturm-Liouville problem with specific boundary conditions.

$$\nabla^2\phi_{ilq}(x, y, z) + \lambda_{ilq}^2\phi_{ilq}(x, y, z) = 0;\ (x, y, z) \in D \tag{8}$$

$$\left.\frac{\partial\phi_{ilq}(x, y, z)}{\partial x}\right|_{x=0,L_x} = \left.\frac{\partial\phi_{ilq}(x, y, z)}{\partial y}\right|_{y=0,L_y} = 0 \tag{9}$$

$$\kappa\left.\frac{\partial\phi_{ilq}(x, y, z)}{\partial z}\right|_{z=-L_z} = h_p\phi_{ilq}(x, y, -L_z) \tag{10}$$

$$\kappa\left.\frac{\partial\phi_{ilq}(x, y, z)}{\partial z}\right|_{z=0} = h_s\phi_{ilq}(x, y, 0) \tag{11}$$

The solutions of Sturm-Liouville problem form a set of completely ortho-normal bases in the spatial domain of die, and their general form can be found in [7]. By applying the general form into our problem, setting h_s to be zero (This assumption is only for comparing our method with [6] under the same experimental setting. Our solver can easily take into account the effect of second heat flow path.), and with several manipulations, $\phi_{ilq}(x,y,z)$ can be got as

$$\phi_{ilq}(x,y,z) = \frac{\cos(\frac{i\pi x}{L_x})\cos(\frac{l\pi y}{L_y})\cos(\lambda_{z_q}z)}{\sqrt{N_{ilq}}}, \tag{12}$$

where N_{ilq}=$\theta_{il}L_xL_yN_{z_q}$, θ_{00}=1/2, θ_{i0}=θ_{0l}=1/4, θ_{il}=1/8 with i≠0 and l≠0, N_{z_q}=L_z+$\kappa/h_p\times\sin^2(\lambda_{z_q}L_z)$, and $\kappa/h_p\times\lambda_{z_q}$=cot $\left(\lambda_{z_q}L_z\right)$.

Those solutions, $\phi_{ilq}(x,y,z)$'s, are called eigenfunctions. Each of them has a corresponding non-zero eigenvalue, λ^2_{ilq}, and N_{ilq} is the normalized value. Each eigenvalue is equal to

$$\lambda^2_{ilq} = \lambda^2_{x_i} + \lambda^2_{y_l} + \lambda^2_{z_q}, \tag{13}$$

where $\lambda^2_{x_i}$=$(i\pi/L_x)^2$ and $\lambda^2_{y_l}$=$(l\pi/L_y)^2$. The $\lambda^2_{x_i}$, $\lambda^2_{y_l}$, $\lambda^2_{z_q}$ are eigenvalues in x-, y-, and z-directions, and λ_{z_q} can be solved by applying the Newton-Raphson method [9].

Essentially, the equivalent thermal conductivity of second heat flow path should not be zero. Setting h_s to be zero which is the same as [6] is only for comparing our method with [6] under the same experimental setting. The effect of second heat flow path can be easily taken into account in our solver because the general solution of Sturm-Liouville problem already takes h_s into account. Thus, only the computational formulas of N_{z_q} and λ_{z_q} are needed to be modified, and this will not influence the derivation of computational formula and evaluating algorithm of the temperature in the remaining portion of this work.

B. System Transformation for Time-Varying Coefficients

Since the generated bases are completely ortho-normal in the spatial domain of die, $T(x,y,z,t)$ can be approximated by $\widehat{T}(x,y,z,t)$ as

$$\widehat{T}(x,y,z,t) = \sum_{q=0}^{N_z-1}\sum_{l=0}^{N_y-1}\sum_{i=0}^{N_x-1}\psi_{ilq}(t)\phi_{ilq}(x,y,z), \tag{14}$$

where $\psi_{ilq}(t)$ is the time-varying coefficient, N_x, N_y, and N_z are truncation points in x-, y-, and z-directions, respectively.

Here, our major goal is to find an analytical expression of $\psi_{ilq}(t)$ for achieving an accurate temperature approximation. Substituting equation (14) into (3), the residual function $r(x,y,z,t)$ is equal to

$$r(x,y,z,t) = \kappa\nabla^2\epsilon(x,y,z,t) - \sigma\frac{\partial\epsilon(x,y,z,t)}{\partial t}, \tag{15}$$

where $\epsilon(x,y,z,t) = T(x,y,z,t) - \widehat{T}(x,y,z,t)$.

To accurately approximate $T(x,y,z,t)$ by equation (14), the norm of $r(x,y,z,t)$ should be as small as possible. In order to achieve this goal, the following steps are performed to find the desired expression of $\psi_{ilq}(t)$. Due to the limitation of space, we only list the derived steps of $\psi_{ilq}(t)$ in brief and the detail derivation is ignored.

- Expand $r(x,y,z,t)$ by using $\phi_{ilq}(x,y,z)$ as

$$r(x,y,z,t) = \sum_{q=0}^{\infty}\sum_{l=0}^{\infty}\sum_{i=0}^{\infty}r_{ilq}(t)\phi_{ilq}(x,y,z), \tag{16}$$

  where $r_{ilq}(t)$=$\int_{-L_z}^{0}\int_0^{L_y}\int_0^{L_x} r(x,y,z,t)\phi_{ilq}(x,y,z)dxdydz$.
- Perform the Galerkin's scheme [8] which sets the $r_{ilq}(t)$'s to be zeros up to truncation points N_x, N_y, and N_z.
- Transfer the resulted equation to the form which preserves the law of conservation by using Divergence Theorem [7].
- Apply equation (8) and the ortho-normality of $\phi_{ilq}(x,y,z)$'s to the resulted equation for getting the following un-coupled system.

$$\sigma\psi'_{ilq}(t) = -\kappa\lambda^2_{ilq}\psi_{ilq}(t) + \widehat{p}_{ilq}(t), \text{ and } \psi_{ilq}(0) = 0; \tag{17}$$

  where $\widehat{p}_{ilq}(t) = \int_0^{L_y}\int_0^{L_x} p(x,y,t)\phi_{ilq}(x,y,0)dxdy$, $0 \le i \le N_x$, $0 \le l \le N_y$, and $0 \le q \le N_z$,
- Obtain the general solution of each $\psi_{ilq}(t)$'s as following.

$$\psi_{ilq}(t) = \frac{1}{\sigma}\int_0^t \widehat{p}_{ilq}(\tau)e^{-\frac{k}{\sigma}\lambda^2_{ilq}(t-\tau)}d\tau. \tag{18}$$

Equation (18) can be applied to obtain the steady-state temperature without any time step evaluation, and equation (17) is applied to obtain the transient temperature distribution.

C. Average Temperature Evaluation of Grid Cells

Generally, the hot spots occur in the regions closing to power sources. Hence, we focus on evaluating the average temperature of each grid cell on the top surface (z=0) of die. First, we present the formulation for calculating the average steady-state temperature and analyze its convergent property. Then, the fast evaluating algorithms are developed for realizing the formulation. Finally, the transient analysis is given.

C.1. Steady State Formulation and its Convergence Rate Analysis

When calculating the steady-state temperature distribution, each turning on/off function of grid cell is a unit step function. Hence, the close-form of each $\psi_{ilq}(\infty) = \widehat{p}_{ilq}(\infty)/(k\lambda^2_{ilq})$ can be analytically obtained from equation (18) without any time step evaluation. After that, plugging $\phi_{ilq}(x,y,z)$'s and $\psi_{ilq}(\infty)$'s into equation (14), the average steady state rising temperature, $\overline{T}_{mn}$, of grid cell (m,n) on the top surface is

$$\begin{aligned}\overline{T}_{mn} &= \frac{1}{\Delta x\Delta y}\int_{n\Delta y}^{(n+1)\Delta y}\int_{m\Delta x}^{(m+1)\Delta x}\widehat{T}(x,y,0,\infty)dxdy \\ &= \sum_{l=0}^{N_y-1}\sum_{i=0}^{N_x-1}K_{il}\cos\left(\frac{i\pi(2m+1)}{2M}\right)\cos\left(\frac{l\pi(2n+1)}{2N}\right)\end{aligned} \tag{19}$$

where

$$K_{il} = \frac{\widehat{P}_{il}}{\kappa}\sum_{q=0}^{N_z-1}\frac{C_{ilq}}{N_{ilq}}, \tag{20}$$

$$C_{ilq} = \begin{cases} \frac{\Delta x\Delta y}{\lambda^2_{ilq}}; & i=0, l=0 \\ \frac{4NL_y\Delta x\sin^2\left(\frac{l\pi}{2N}\right)}{l^2\pi^2\lambda^2_{ilq}}; & i=0, l\neq 0 \\ \frac{4ML_x\Delta y\sin^2\left(\frac{i\pi}{2M}\right)}{i^2\pi^2\lambda^2_{ilq}}; & i\neq 0, l=0 \\ \frac{16MNL_xL_y\sin^2\left(\frac{i\pi}{2M}\right)\sin^2\left(\frac{l\pi}{2N}\right)}{i^2l^2\pi^4\lambda^2_{ilq}}; & i\neq 0, l\neq 0 \end{cases} \tag{21}$$

and

$$\widehat{P}_{il} = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1}p_{mn}\cos\left(\frac{i\pi(2m+1)}{2M}\right)\cos\left(\frac{l\pi(2n+1)}{2N}\right). \tag{22}$$

Before introducing our evaluating algorithms for calculating the average steady state rising temperature, we first proceed the convergent analysis to show the benefit of our temperature calculating formula. The error of our temperature calculating formula can be bounded by the following theorem.

Theorem 1: The absolute truncation error of average steady state temperature for each grid cell (m,n) by using the GIT based formulation with truncation points N_x, N_y, and N_z in x-, y-, and z-directions is bounded by

$$\sum_{(i,l,q)\in S_1}\frac{\alpha_1}{i^2l^2\lambda^2_{ilq}} + \sum_{(i,q)\in S_2}\frac{\alpha_2}{i^2\lambda^2_{i0q}} + \sum_{(l,q)\in S_3}\frac{\alpha_3}{l^2\lambda^2_{0lq}} + \sum_{q\in S_4}\frac{\alpha_4}{\lambda^2_{00q}}, \tag{23}$$

where S_1=$[1,N_x]\times(N_y,\infty)\times[0,\infty)\cup(N_x,\infty)\times[1,N_y]\times[0,\infty)\cup[1,N_x]\times[1,N_y]\times(N_z,\infty)$, S_2=$[1,N_y]\times(N_z,\infty)\cup(N_x,\infty)\times[0,\infty)$, S_3=$[1,N_x]\times(N_z,\infty)\cup(N_y,\infty)\times[0,\infty)$, S_4=(N_z,∞), α_1=$128M^2N^2P_T/(L_xL_yL_z\kappa\pi^4)$, α_2=$16M^2P_T/(L_xL_yL_z\kappa\pi^2)$, α_3=$16N^2P_T/(L_xL_yL_z\kappa\pi^2)$, *and* α_4=$2P_T/(L_xL_yL_z\kappa)$.

Due to the limitation of space, the proof is ignored. The above result shows that the decaying rate of the truncation error of our GIT based method can be in the order of $i^2l^2((i\pi/L_x)^2+(l\pi/L_y)^2+\lambda_{z_q}^2)$.

To compare the convergent rate, we also obtain the truncation error bound of formulation in [6] as following.

$$\sum_{(i,l)\in B_1}\frac{\beta_1}{i^2l^2\gamma_{il}}+\sum_{i\in B_2,l=0}\frac{\beta_2}{i^2\gamma_{il}}+\sum_{i=0,l\in B_3}\frac{\beta_3}{l^2\gamma_{il}}, \qquad (24)$$

where γ_{il}=$\sqrt{(i\pi/L_x)^2+(l\pi/L_y)^2}$, B_1=$(N_x,\infty)\times(N_y,\infty)$, $B_2=(N_x,\infty)$, B_3=(N_y,∞), β_1=$64M^2N^2P_T/(L_xL_y\kappa\pi^4)$, $\beta_2=8M^2P_T/(L_xL_y\kappa\pi^2)$, and $\beta_3=8N^2P_T/(L_xL_y\kappa\pi^2)$.

This bound shows that the decaying rate of the truncation error of Green's function based formulation is in the order of $i^2l^2\sqrt{(i\pi/L_x)^2+(l\pi/L_y)^2}$.

Therefore, the convergence rate of GIT based method is much faster than Green's function based method [6]. The reason is that the GIT based method generates the ortho-normal spatial bases for the *transient heat diffusion equation*, and obtains the close-form of steady state solution by using these spatial bases which can fully fill the eigen-space of heat diffusion equation. On the other hand, [6] constructs the spatial approximated function by applying Green's function to Possion's equation which does not contain the temporal information. As a result, the generated Green's function could not fully fill the eigen-space of transient heat diffusion equation for approximating the temperature.

The convergence rate of our GIT based formulation is not only faster than [6], the experimental results also demonstrate that it can maintain the same accuracy as [6] even if the truncation point, N_x or N_y, is far less than the number of divisions, M or N.

Although the truncation points N_xN_y can be far less than the number of grid cells MN, there is no actual efficiency improvement over [6] if we directly apply the standard FFT to evaluate each T_{mn} because the standard FFT need pad zeros to the input data when the dimensions of input and output data are not equal. To overcome this limitation, we provide fast evaluating algorithms for our GIT formulation without the zero padding.

C.2. Fast Evaluating Algorithms for GIT Formulation

To efficiently realize our formulation of steady state temperature distribution, we first derive a one-dimensional radix-two based FFT like evaluating algorithm for the length of output data being larger than the length of input data, *1D-STL-FFT*. Then, based on *1D-STL-FFT*, we develop another one-dimensional FFT like evaluating algorithm for the length of output data being smaller than the length of input data, *1D-LTS-FFT*. Finally, these two algorithms are integrated to calculate equations (19) and (22) by the row-column procedure, and the computational complexity of our GIT based thermal simulator can be analyzed to be only $\underline{O(MN\log_2 N_xN_y)}$.

a) 1D-STL-FFT: The prototype of *1D-STL-FFT* is

$$\overline{F}_k=\sum_{i=0}^{N_x-1}f_ie^{j2\pi ik/2M};\quad k=0,\cdots,2M-1, \qquad (25)$$

where N_x<M and each is power of 2, j=$\sqrt{-1}$, and f_i's and $\overline{F}_k$'s are complex input and output data, respectively. Since the length of $\overline{F}_k$'s is larger than the length of f_i's, the zeros-padding step of f_i's used in standard FFT algorithm for evaluating $\overline{F}_k$'s should be avoided to save runtime. Therefore, our *1D-STL-FFT* algorithm, a modified FFT algorithm, is developed as stated in Fig 2 to calculate Equation (25) without zeros-padding. Firstly, the "**Reverse-bit**(f)" performs $\log_2 N_x$ times of the Danielson-Lanczos lemma [9] to Equation (25) and reorders the input data. Then, the *1D-STL-FFT* algorithm evaluates the output of those L sub Discrete Fourier Transforms (DFTs) in the bottom level by using *Line* 3∼16, and performs *Line* 17 to get the output of rest levels.

```
Algorithm Radix-two 1D-STL-FFT
Input: Complex vector $f$ with length $N_x$
Output: Complex vector $\overline{F}$ with length $2M$
1     Begin
2       $f_{\mathbf{R}}$ = Reverse-bit($f$) ;
3       $L = 4M/N_x$ ;
4       $N_{SubDFTs} = N_x/2$ ;
5       For $SubIndex$ = 0 to $N_{SubDFTs} - 1$
6           $k = L \times SubIndex$ ;
7           $i = 2 \times SubIndex$ ;
8           For $SubK$ = 0 to $L - 1$
9               $\overline{F}[k] = \overline{F}[k] + f_{\mathbf{R}}[i] + f_{\mathbf{R}}[i+1] \times e^{j2\pi\times SubK/L}$ ;
10              $k = k + 1$ ;
11          EndFor
12      EndFor
13      Apply the bottom up procedure of standard FFT to
        execute $\log_2 N_x - 1$ times Danielson-Lanczos lemma
        of (25) for evaluating the final $\overline{F}$
14    End
```

Fig. 2. Procedure of *1D-STL-FFT*. The "**Reverse-bit**" means the reverse-bit algorithm [9]

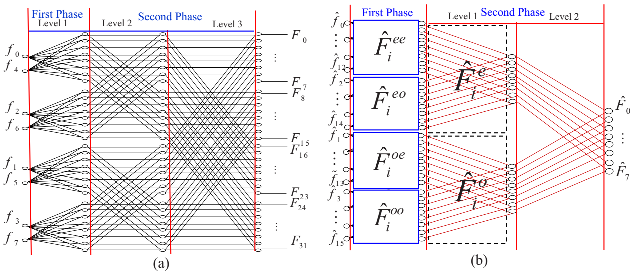


Fig. 3. Computational flow graphs of *1D-STL-FFT* and *1D-LTS-FFT* with $N_x=8$ and $M=16$. (a) The *1D-STL-FFT*. (b) The *1D-LTS-FFT*.

An example with $M=16$ and $N_x=8$ is given in Fig. 3.(a), there are 3 bisecting levels, and 4 sub DFTs in the bottom level. After performing the reverse-bit algorithm to input data, two phases are executed. The first phase is done by using *Line* 3∼16 of Fig. 2. Then, the second phase is to get output of rest levels by executing the bottom up procedure of standard FFT as stated in *Line* 17 of Fig. 2.

The complexity of *1D-STL-FFT* is $O(M\log_2 N_x)$ because there are $\log_2 N_x$ bisecting levels and the complexity of each level is $O(M)$.

b) 1D-LTS-FFT: The prototype of *1D-LTS-FFT* is

$$\widehat{F}_i=\sum_{m=0}^{M-1}\widehat{f}_me^{j2\pi im/2M};\quad i=0,\cdots,N_x-1, \qquad (26)$$

where $M>N_x$, and $\widehat{F}_i$ and $\widehat{f}_m$ are complex output and real input data, respectively. Repeating the Danielson-Lanczos lemma with $\log_2(M/N_x)+1$ times, $\widehat{F}_i$ can be written as the sum of $2M/N_x$ sub DFTs, and each sub DFT has the same form as the *1D-STL-FFT* with input and output length are $N_x/2$ and N_x, respectively.

Two phases are utilized to evaluate $\widehat{F}_i$ and the *1D-LTS-FFT* algorithm is summarized in Fig. 4. First, *Line* 2 performs the reverse-bit algorithm to the input data, and *Line* 4∼8 use the *1D-STL-FFT* algorithm to obtain each bisected sub DFT. After each sub DFT being done, a bottom up procedure is applied to the rest $\log_2(M/N_x)+1$ bisecting levels for finding $\widehat{F}_i$ and the executing steps are from *Line* 9∼26.

```
Algorithm Radix-two 1D-LTS-FFT
Input: Real vector $\widehat{f}$ with length $M$
Output: Complex vector $\widehat{F}$ with length $N_x$
1  Begin
2      $\widehat{f}_{\mathbf{R}}$ = Reverse-bit($\widehat{f}$) ;
3      $N_{SubDFTs} = 2M/N_x$ ;
4      For $Sub_i$ = 0 to $N_{SubDFTs} - 1$
5          $Start$ = $Sub_i \times N_x$ ;
6          $End$ = $Start + N_x$;
7          $F_t(Start : End - 1)$ = 1D-LTS-FFT$\Big(\widehat{f}_{\mathbf{R}}(\frac{Start}{2} : \frac{End}{2} - 1)\Big)$ ;
8      EndFor
9      $L$ = $N_x$ ;
10     For $level$ = 0 to $log_2(M/N_x)$
11         $Next^*$ = 0 ;
12         $Sub_i$ = 0 ;
13         $N_{SubDFTs}$ = $N_{SubDFTs}$ / 2 ;
14         While $Sub_i < N_{SubDFTs}$
15             For $i$ = 0 to $N_x - 1$ ;
16                 $b1$ = $i$ + $Sub_i \times N_x$ ;
17                 $b2$ = $b2$ + $N_x$ ;
18                 $n$ = $i$ + $Next^*$ ;
19                 $F_t[n]$ =$F_t[b1]$+$F_t[b2]\times$ $e^{j2\pi i/L}$ ;
20             EndFor
21             $Sub_i$ = $Sub_i$ + 2 ;
22             $Next^*$ = $Next^*$ + $N_x$ ;
23         EndWhile
24         $L$ = 2 $\times$ $L$ ;
25     EndFor
26     $\widehat{F}$ = $F_t(0 : N_x - 1)$;
27 End
```

Fig. 4. Procedure of *1D-LTS-FFT*.

An example with $M = 16$ and $N_x = 8$ is shown in Fig. 3.(b). In the first phase, the input data are reordered by using the reverse-bit algorithm, and these reordered data are fed into the corresponding *1D-STL-FFT* blocks. This can be done by using *Line* 3∼8 in Fig. 4. Then, the output of top block in the level 1 of second phase is calculated by

$$\widehat{F}_i^e = \widehat{F}_i^{ee} + e^{j2\pi i/16}\widehat{F}_i^{eo}, \tag{27}$$

and $\widehat{F}_i^o$ can be calculated by using a similar way. Finally, $\widehat{F}_i$ is equal to

$$\widehat{F}_i = \widehat{F}_i^e + e^{j2\pi i/32}\widehat{F}_i^o. \tag{28}$$

The above computational flow of the second phase is summarized from *Line* 9∼26 in Fig. 4.

For the general case, the sub DFTs in each level of the second phase can be obtained by combining those sub DFTs of their previous level with the similar formula of equation (27) by replacing 16 to be $2N_x$, $4N_x$, $\cdots$, $2M$ in each level. The computational complexity of first phase is $O(M\log_2 N_x)$ because the *1D-STL-FFT* need to be executed $2M/N_x$ times, and each complexity is $O(N_x \log_2 N_x)$. The complexity is $O(M)$ for the second phase. Hence, the computational complexity of *1D-LTS-FFT* is $O(M\log_2 N_x)$.

c) Temperature Evaluation: The average steady state rising temperature, $\overline{T}_{mn}$, shown in equation (19), can be evaluated as

$$\overline{T}_{mn} = \frac{1}{2}R_e\left\{\overline{F}_{m,n} + \overline{F}_{2M-(m+1),n}\right\}, \tag{29}$$

where $R_e\left\{\cdot\right\}$ is the real part operator, and

$$\overline{F}_{k_1,k_2} = \sum_{i=0}^{N_x-1}\sum_{l=0}^{N_y-1}\overline{K}_{il}e^{\frac{j2\pi ik_1}{2M}}e^{\frac{j2\pi lk_2}{2N}}. \tag{30}$$

Here, $0\leq k_1 \leq 2M-1$, $0\leq k_2 \leq 2N-1$, $\overline{K}_{il}$=$K_{il}e^{j2\pi i/4M}e^{j2\pi l/4N}$, and each K_{il} is equal to equation (20).

In the following, we are going to utilize the *1D-STL-FFT* algorithm to develop a row-column procedure to calculate $\overline{F}_{k_1,k_2}$'s. The K_{il}'s can also be obtained by using a similar procedure with the *1D-LTS-FFT* algorithm. The row-column based *2D-STL-FFT* method for calculating $\overline{F}_{k_1,k_2}$'s is summarized in Fig. 5. *Line* 2∼4 performs *1D-STL-FFT* to each row of the input matrix $\overline{K}$ which each (i,l) entry is equal to $\overline{K}_{il}$, and then *Line* 5∼7 applies *1D-STL-FFT* to each column of output matrix got from the row procedure to obtain the desire matrix $\overline{F}$. Since the complexity of *1D-STL-FFT* is $O(M\log_2 N_x)$, the total complexity of evaluating $\overline{F}_{k_1,k_2}$'s by this row-column procedure is $\underline{O(MN\log_2 N_x)}$.

```
Algorithm Radix-two 2D-STL-FFT
Input: Complex matrix $\overline{K}$ with length $N_x \times N_y$
Output: Complex matrix $\overline{F}$ with length $2M\times 2N$
1      Begin
2          For i = 0 to $N_x - 1$
3              $T_{Row}(i, 0 : 2N - 1)$ = 1D-STL-FFT$\Big(\overline{K}(i, 0 : N_y - 1)\Big)$ ;
4          EndFor
5          For j = 0 to $2N - 1$
6              $\overline{F}(0 : 2M - 1, j)$ = 1D-STL-FFT($T_{Row}(0 : N_x - 1, j)$) ;
7          EndFor
8      End
```

Fig. 5. Procedure of *2D-STL-FFT*.

To obtain each K_{il} from equation (20), $\widehat{P}_{il}$'s need to be known from equation (22). Therefore, the two dimensional type of equation (26) is needed to get related $\widehat{F}_{i,l}$'s for input data being p_{mn}'s. Similarly, a *1D-STL-FFT* based row-column procedure can be used to get those related $\widehat{F}_{i,l}$'s. However, the form of equation (29) can not be utilized to calculate $\widehat{P}_{il}$'s because the lengths of those related $\widehat{F}_{i,l}$'s in row and column directions are less than $2M$ and $2N$, respectively. Therefore, the complex conjugates of $\widehat{F}_{i,l}$'s are required to complete the calculation of $\widehat{P}_{il}$'s.

Fortunately, the complex conjugate of the output from each sub *1D-STL-FFT* in calculating $\widehat{F}_{i,l}$'s can be directly obtained by reversing these sub DFTs , for example, $(\widehat{F}_i^{ee})^*$=$\widehat{F}_{N_x-i}^{ee}$ in Fig. 4. Therefore, the complex conjugate of $\widehat{F}_{i,l}$'s can be got by firstly reversing the data of F_t from *Line* 7 in Fig. 4, and performing *Line* 9∼26 in Fig. 4 during the row-column procedure of $\widehat{F}_{i,l}$'s.

Similar to the analysis of $\overline{F}_{k_1,k_2}$'s, the complexity for evaluating $\widehat{F}_{i,l}$'s is $O(MN\log_2 N_y)$. The complexity of calculating the negative frequency components of $\widehat{F}_{i,l}$'s is $O(MN) + O(N_yN)$ since only the second phase need to be recomputed. Therefore, the complexity for computing equation (22) is $\underline{O(MN\log_2 N_y)}$.

¿From the above discussion, we conclude that the complexity of our GIT based thermal simulator is $\underline{O(MN\log_2 N_xN_y)}$.

C.3. Transient Simulation

To perform the transient simulation, the turning on/off function of each grid, $H_{mn}(t)$, is a time interval function specified by instruction. After applying finite difference schemes (For simplicity, we use the backward-Euler method.) to equation (17), each time-varying coefficient, ψ_{ilq}^t, at the sampling time t is

$$\psi_{ilq}^t = \frac{\sigma}{R_{ilq}}\psi_{ilq}^{t-\Delta t} + \frac{\Delta t}{R_{ilq}\sqrt{N_{ilq}}}\widehat{P}_{il}^t, \tag{31}$$

where Δt is the time step, $R_{ilq} = \sigma + \kappa\lambda_{ilq}^2\Delta t$, $\widehat{P}_{il}^t$ is equal to equation (22) with p_{mn} replaced by $p_{mn}H_{mn}^t$, and H_{mn}^t is the value of turning on/off function at time step t. After time-varying coefficients at time t being calculated, the average temperature in each grid cell at time step t can be obtained by equation (14) with the same evaluating method presented in previous subsection.

IV. Experimental Results

We implement our GIT based thermal simulator and the Algorithm II of a highly efficient Green's function based method [6] in C++ language. The state-of-the-art FFT package, FFTW3 [10], is used to realize the DCT and IDCT for [6]. All methods are tested on a HP xw9300 workstation with 16 GB memory. The results are compared with a commercial computational fluid dynamic software, ANSYS®.

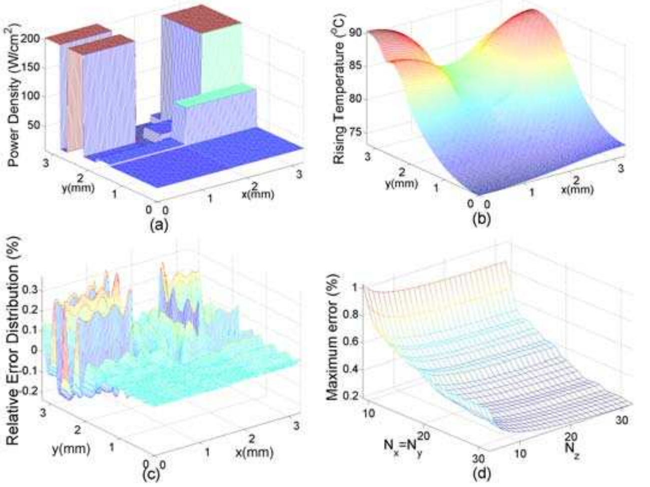


Fig. 6. Accuracy and the maximum error trend of proposed GIT based method. (a) The power density map of test chip. (b) The estimated temperature distribution of test chip. (c) The relative error distribution of GIT based method compared with the result of ANSYS®. Here, the numbers of truncation points are 16×16×70. (d) The maximum relative error versus the numbers of truncation points.

A. Accuracy and Fast Convergence of the GIT Based Thermal Simulator

The chip, DEC Alpha 21264, is employed to demonstrate the accuracy of our method, and its size is scaled down to 3.3mm×3.3mm×0.5mm. Its power density map is shown in Fig. 6.(a), and the power sources are on the top surface of functional blocks. The equivalent heat transfer coefficient of primary heat transfer path is 8700 W/($\mathrm{m}^2{\cdot}^\circ\mathrm{C}$), and the thermal conductivity of silicon is 148 W/($\mathrm{m}{\cdot}^\circ\mathrm{C}$). The above settings are the same as [6]. The power density map is divided into 128×128 grid cells. The average steady state temperature distribution on the top surface of die computed by our GIT based method with the truncation point being 16 in each x-, y-direction, and 70 in z-direction is shown in Fig. 6.(b). The maximum relative error compared with the result of ANSYS® is 0.3732%, and its relative error distribution is shown in Fig. 6.(c). On the other hand, the truncation point of Green's function based method [6] need to be 2048 in both x- and y-directions that its number of bases is 234 times larger than our method to achieve the same accuracy level, and its maximum relative error is 0.3735%. This reveals the fast convergence advantage of our proposed GIT based method. To further demonstrate our fast convergence rate, we plot the maximum relative errors with different truncation points in Fig 6.(d). As you can see that our GIT based analyzer can achieve an extremely accurate solution even when the truncation points are very small.

B. Thermal Simulation for Full-Chip Containing Lots of Functional Blocks

To demonstrate the capability of our GIT based method for the thermal simulation of full-chip with containing lots of functional blocks, and the efficiency improvement of our GIT based method over the Algorithm II of Green's function based method [6], we consider a test chip with dimension 1cm×1cm×0.5mm. It consists of one million functional blocks, the power density of each block is between 3.0e4 W/m^2 and 1.5e6 W/m^2, and its power density map is illustrated in Fig. 7.(a). The top surface of this chip is discretized into 1024×1024 square grid cells. The truncation points of proposed method are set to be 16×16×8 and the truncation points of [6] are set to be 2048×2048 to achieve the similar maximum error. The temperature distribution of top surface calculated by the proposed method is shown in Fig. 7.(b), and its maximum error is only 0.3576% presented in TABLE I.

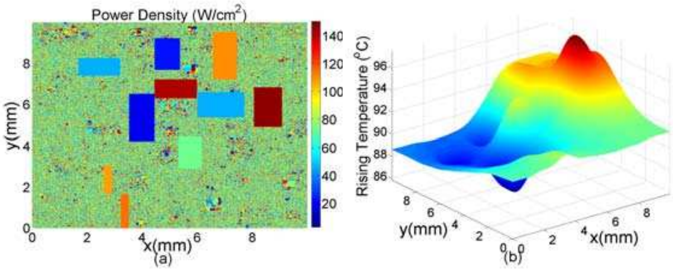


Fig. 7. The power density and temperature distribution of a 1cm×1cm chip with one million functional blocks. (a) The power density map. (b) The estimated temperature distribution.

The runtime comparison is shown in TABLE I. The run-time of post-calculating stage in our method is only 0.1312 seconds while the run-time of post-calculating stage in [6] is 2.7642 seconds. The speedup of our method over the Algorithm II of [6] is 21.07 at the post-calculating stage. This result demonstrates the dramatic efficiency improvement of our thermal analyzer over [6].

		Green's function based [6]	GIT based
number of functional blocks		1 million	
number of grid cells		2^{20}	
number of bases		2^{22}	2^{11}
max error (%)		0.4143	0.3576
runtime (s)	pre-calculating	2.4785	0.00005
	post-calculating	2.7642	0.1312
speedup (post-calculating)		21.0686	

TABLE I
COMPARISON OF THE PROPOSED GIT BASED METHOD AND ALGORITHM II OF [6] FOR A CIRCUIT WITH ONE MILLION FUNCTIONAL BLOCKS.

V. CONCLUSIONS

An accurate and efficient GIT based thermal simulator has been presented. Experimental results confirm its theoretical property which can achieve extremely accurate results with sufficiently small truncation points. The proposed algorithm only takes 0.13 seconds for the thermal analysis of full chip with one million functional blocks and over one million grid cells in the post-calculating stage to achieve accurate steady state temperature distribution. Therefore, the proposed GIT based thermal simulator is very suitable for the thermal-aware design flow. Finally, the early-stage 3-D chip thermal analysis can be achieved by numerical schemes and combining our proposed analytical technique in this paper, and this will be our future work.

REFERENCES

[1] T. -Y. Wang and C. C. -P. Chen, "Thermal-ADI: A Linear-Time Chip-Level Thermal Simulation Algorithm Based on Alternating-Direction Implicit (ADI) Method," in *TVLSI*, vol. 11, no. 4, pp. 691-700, Aug. 2003.

[2] T. -Y. Wang and C. C. -P. Chen, "SPICE-Compatible Thermal Simulation with Lumped Circuit Modeling for Thermal Reliability Analysis Based on Model Reduction," in *ISQED*, pp. 357-62, Mar. 2004.

[3] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "IC Thermal Simulation and Modeling via Efficient Multigrid-Based Approaches," in *TCAD*, vol. 25, no. 9, pp. 319-26, Sep. 2006.

[4] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron and M. R. Stan "HotSpot:ACompact Thermal Modeling Methodology for Early-Stage VLSI Design," in *TVLSI*, vol. 14, no. 5, pp. 501-13, May 2006.

[5] J.-L. Tsai, C. C.-P. Chen, G. Chen, B. Goplen, H. Qian, Y. Zhan, S.-M. Kang, M. D. F. Wong and S. S. Sapatnekar, "Temperature-Aware Placement for SOCs," in *Proceedings of the IEEE*, vol. 94, no. 8, pp. 1502-18, Aug. 2006.

[6] Y. Zhan, and S. S. Sapatnekar, "High Efficiency Green Function-Based Thermal Simulation Algorithms," in *TCAD*, accepted for future publication.

[7] M. D. Mikhailov, and M. N. Ozisik, "Unified Analysis and Solutions of Heat and Mass Diffusion," John Wiley & Sons Inc., NY, 1983.

[8] M. D. Mikhailov, "General Solutions of the Diffusion Equations Coupled at the Boundary Conditions," in *Int. J. Heat Mass Transf.*, vol. 16, pp. 2155-64, 1973.

[9] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, "Numerical Recipes in C++," Cambridge Unvi. Press, 2002.

[10] M. Frigo and S. G. Johnson, "FFTW version 3.1 package," in http://www.fftw.org.