High Efficiency Green Function-Based Thermal Simulation Algorithms

Yong Zhan and Sachin S. Sapatnekar, *Fellow, IEEE*

***Abstract*— Due to technology scaling trends, the accurate and efficient calculations of the temperature distribution corresponding to a specific circuit layout and power density distribution will become indispensable in the design of high performance VLSI circuits. In this paper, we present three highly efficient thermal simulation algorithms for calculating the on-chip temperature distribution in a multilayered substrate structure. All three algorithms are based on the concept of the Green function and utilize the technique of discrete cosine transform (DCT). However, the application areas of the algorithms are different.The first algorithm is suitable for localized analysis in thermal problems, while the second algorithm targets full-chip temperature profiling. The third algorithm, which combines the advantages of the first two algorithms, can be used to perform thermal simulations where the accuracy requirement differs from place to place over the same chip. Experimental results show that all three algorithms can achieve relative errors of around 1% compared with that of a commercial computational fluid dynamic (CFD) software package for thermal analysis while their efficiencies are orders of magnitude higher than that of the direct application of the Green function method.**

***Index Terms*— Simulation, thermal analysis, multilayered substrate, Green function method, discrete cosine transform (DCT), table look-up approach, spectral domain analysis.**

I. Introduction

As the electronics market continues pushing forward the performance of VLSI circuits, the escalating power consumption has become a severe problem in chip design. Higher power consumption leads to elevated on-chip temperature, which consequently affects both the performance and reliability of circuits. In [1], the authors pointed out that the delay of aluminum interconnect goes up by 30% when the temperature rises from 25°C to 100°C, and in [2], it was reported that the electromigration-induced mean-time-to-failure of interconnect is reduced by 90% when the temperature increases from 25°C to 52.5°C. This situation has made it imperative to incorporate thermal effects into physical design tools for chip design so as to accelerate the design closure and improve the quality of the final product.

The first step towards the development of a thermal-aware physical design tool is to obtain the capability of calculating the on-chip temperature distribution accurately and efficiently given a power density distribution. The efficiency of the temperature-calculating algorithm is of paramount importance especially in early stages of physical design such as thermal-aware floorplanning and placement, since for these design steps, thermal analysis is often used as part of the simulation core of an optimization engine where a large design space of possible physical layouts must be explored and an independent calculation on temperature distribution has to be performed for each candidate layout.

Based on the type of analysis they perform, thermal simulation algorithms can be generally divided into two categories, i.e., those for transient analysis and those for steady-state analysis. Transient analysis is concerned with the evolution of temperature distribution within a chip given a time-varying power density distribution, and can be performed efficiently using the thermal ADI algorithm proposed by Wang *et al.* in [3]. Steady-state analysis, on the other hand, is interested in the stabilized temperature distribution given a time-independent power density distribution or a power density distribution averaged over time. In this paper, we will focus on the steady-state thermal analysis.

Several steady-state thermal simulation algorithms have been used previously in chip design. The finite difference method (FDM) [4] and the finite element method (FEM) [5] obtain the temperature distribution through meshing the silicon substrate and solving a system of linear equations relating the temperatures of grid cells to the power density distribution. The difference between the two methods is that the FDM discretizes the differential operator of the governing equation of thermal effects, while the FEM discretizes the field. The advantages of the FDM and FEM include their robustness and high accuracy. In addition, the FEM also possesses the capability of handling complicated boundary conditions. The primary drawback of the FDM and FEM rests on the fact that they always require volume meshing of the entire substrate even though the devices are usually fabricated only in a thin layer close to the top surface of the IC chip. Hence, even for the cases where only the temperature distribution within the device layer is of interest, we still have to solve a large system of linear equations corresponding to the volume meshing, which leads to low efficiency. In [6], a thermal simulation algorithm based on the solution of the finite difference equations using the multigrid approach was proposed, and its high efficiency has made the full-chip thermal simulation practical for the optimizations in physical designs.

The boundary element method (BEM) constitutes another class of thermal simulation algorithms in which the volume meshing of the substrate is avoided. An important underlying concept in the BEM is the Green function, which describes the temperature distribution within the chip when a unit point power source is present. For the simple geometries encountered in chip design, the explicit form of the Green function can be obtained, and the temperature field under an arbitrary power density distribution can be calculated by integrating the corresponding Green function. Because the BEM only meshes the power generating

This work was supported in part by the National Science Foundation under awards CCF-0541367 and CCF-0205227, by the Semiconductor Research Corporation under contract 2003-TJ-1092, and by DARPA under grant N66001-04-1-8909 from the US Navy. The authors thank the University of Minnesota Supercomputing Institute for the use of their computing facilities.

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (email: yongzhan@ece.umn.edu; sachin@ece.umn.edu).

surfaces in thermal simulations as opposed to the meshing of the entire substrate by the FDM and FEM, it naturally leads to a smaller problem size, and hence has the potential of achieving high efficiency. However, the actual runtime of an algorithm implemented using the BEM depends critically on how efficient the Green function is evaluated and how the temperature distribution is calculated given the power density distribution. In [7], the classical Green function approach was used in thermal simulations where the Green function was utilized directly to evaluate the temperature field in a rectangular-shaped substrate. Because the underlying Green function is expressed as a multiple-infinite summation and it has to be truncated at high indices in actual implementations to maintain a reasonable accuracy, the efficiency of this method is rather low. In [8], the method of images was used to obtain the Green function in closed form at the expense of relaxing the boundary conditions by assuming that the chip is infinitely large horizontally. The advantage of this method is that the Green function can be computed on-line efficiently and thus it is suitable for optimization purposes. However, by assuming that the chip is infinitely large horizontally, the on-chip temperature will be severely under-estimated especially near the boundaries of the actual chip, although the locations of the hot spots can be correctly identified as shown in [8]. In [9], an efficient algorithm for evaluating the temperature field in VLSI chips using a semi-analytical form of the Green function was proposed which takes into account the multilayered nature of the semiconductor substrates used in IC fabrications. Nevertheless, this method also assumes that the chip is infinitely large horizontally, and therefore it has the same problem as [8].

Note that the computation of the steady-state temperature distribution T in thermal problems is very similar to the computation of the potential field ϕ in electrical problems. Both T and ϕ satisfy the Poisson's equation, and the power source P in thermal problems corresponds to the charge q in electrical problems. In [10] and [11], the discrete cosine transform (DCT) is combined with a table look-up approach to improve the efficiency of using the Green function to calculate the electrical potential distribution within a rectangular-shaped substrate. In this method, the multiple-infinite summation contained in the expression of the Green function is not evaluated on-line. Instead, look-up table and vectors are established in advance so that each evaluation of the Green function is reduced to the summation of a constant and 80 terms in the look-up table and vectors. This is a significant improvement over the direct evaluation of the multiple-infinite summation in the classical Green function method, which may involve thousands or even more terms to ensure a reasonable accuracy. Since the look-up table and vectors only have to be computed once for each technology and substrate geometry, but are independent of where the devices are located on the chip, they can be obtained in the pre-characterization phase of the design and used many times in the optimization process. As a result, the amortized cost of establishing the look-up table and vectors can be ignored in practice. Our first thermal simulation algorithm (Algorithm I) follows a similar line of analysis as in [10] and [11]. The difference is that since the boundary conditions encountered in thermal problems are different from those in electrical problems, the Green function and the look-up table and vectors must be re-derived to reflect the special characteristics of the thermal problems.

The improvement in efficiency of Algorithm I, as compared with that of the classical Green function method, comes from its faster evaluation of the expressions involving the Green function in calculating the temperature field, and compared with other fast algorithms such as the ones presented in [8] and [9], our algorithm can achieve a much higher accuracy because it does not assume that the chip is infinitely large horizontally, and hence it can take the proper boundary conditions into consideration. Asymptotically, however, the classical Green function method, the algorithms in [8] and [9], and our Algorithm I all have the same time complexity of $O(\mathcal{N}_s \cdot \mathcal{N}_f)$, where $\mathcal{N}_s$ is the number of power source regions and $\mathcal{N}_f$ is the number of temperature field regions. For cell level full-chip thermal simulations where the number of heat sources is large and the temperature profile over the entire chip is sought, however, a still faster algorithm is required.

In [12], Costa *et al.* proposed an elegant algorithm for efficiently performing the full-chip electrical potential profiling, which is a key step in solving substrate parasitic extraction problems. This algorithm combines the concept of functional eigen-decomposition with the technique of the DCT to reduce the overall runtime of full-chip electrical potential profiling from $O(\mathcal{N}_{gc}^2)$ to $O(\mathcal{N}_{gc} \times \log(\mathcal{N}_{gc}))$, where $\mathcal{N}_{gc}$ is the total number of grid cells. Because of the parallelism between the thermal problem and the electrical problem, we can use a similar approach to reduce the asymptotic runtime of full-chip temperature profiling. We have implemented such an approach in our second algorithm (Algorithm II), and we will present it from the perspective of spectral domain computations that are familiar to engineers. Note that the temperature distribution can be obtained by convolving the power density distribution with the underlying Green function, and it is well known that convolutions in the space domain correspond to point-wise multiplications in the spectral domain. Therefore, using the spectral domain computations in conjunction with the DCT for transforming the data between space and spectral domains, we will be able to significantly reduce the runtime of full-chip temperature profiling. Our algorithm takes a piece-wise constant power density map as the input and generates a piece-wise constant temperature map as the output. The primary steps of the algorithm include:

1) Obtaining the spectral domain representation of the power density map using the 2D DCT. The order of the DCT expansion is determined dynamically by the power density map instead of being set *a priori* to ensure the accuracy.
2) Calculating the spectral domain representation of the temperature map by multiplying each spectral component of the power density map by the corresponding spectral response of the linear system determined by the Green function.
3) Using a 2D inverse discrete cosine transform (IDCT) to obtain the temperature map from its spectral domain representation.

Both the 2D DCT and the 2D IDCT can be calculated efficiently

using the 2D fast Fourier transform (FFT). The asymptotic time complexity of the overall algorithm is $O(\mathcal{N}_{gs}\times\log(\mathcal{N}_{gs})) + O(\mathcal{N}_{gf}\times\log(\mathcal{N}_{gf}))$, where $\mathcal{N}_{gs}$ and $\mathcal{N}_{gf}$ are the number of grid cells in the power source layer and temperature field layer, respectively. Hence, for calculating the full-chip temperature profile, the time complexity of Algorithm II is much smaller than that of Algorithm I, which is $O(\mathcal{N}_{gs}\cdot\mathcal{N}_{gf})$. Note that the lower asymptotic time complexity of Algorithm II does not invalidate the usefulness of Algorithm I because, as will be elaborated in Section III.D, Algorithm I often works better for localized analysis, where the effects of a few critical circuit blocks on the temperature distribution in a few key regions are of interest.

Our third algorithm (Algorithm III) is a combination of Algorithm I and II, and it possesses the capability of performing thermal simulations where the accuracy requirement differs from place to place over the same chip, e.g., in mixed signal designs where analog circuits are fabricated on the same chip as digital circuits, the analog blocks often have more stringent accuracy requirements on thermal simulations because the operations of the analog circuits are more sensitive to temperature. Algorithm III reflects the idea of the pre-corrected FFT, which has been used extensively in the IC parasitic extraction works [13] [14] [15]. The algorithm first uses coarse grids to divide the source and field planes where each grid cell in the source plane can contain several logic gates or analog functional units, and the size of each grid cell in the field plane satisfies the accuracy requirements of the digital circuits. The power density of each grid cell in the source plane can be obtained by adding up the contributions from the logic gates and analog functional units that are located in it. A coarse temperature map for the field plane is then obtained from the coarse power density map using Algorithm II and is used for the digital blocks. Finally, for each analog functional unit on the field plane whose temperature is to be calculated more accurately, we use Algorithm I to compute the contributions to its temperature rise from the nearby logic gates and analog function units on the source plane, and use this result to correct the temperature obtained by Algorithm II over the coarse grid cell.

Our algorithms are all implemented in C++ and experimental results show that they can achieve relative errors of around 1% compared with that of a commercial computational fluid dynamic (CFD) software package for thermal analysis, while their efficiencies are orders of magnitude higher than that of the classical Green function method. The rest of the paper will be organized as follows. In Section II, we formulate the temperature field computation problem and present the concept of Green function for thermal problems. In Section III, we discuss in detail the three thermal simulation algorithms. Section VI shows the experimental results, and the conclusions are provided in Section V.

II. Problem Formulation and the Green Function for Thermal Problems

A. Problem formulation

Fig. 1(a) shows an IC chip with the associated packaging, and Fig. 1(b) shows a schematic of the structure in Fig. 1(a) where the packaging including the heat spreader and the heat sink has been simplified but the multilayered structure of the chip is explicitly shown. The steady-state temperature distribution inside the chip is governed by Poisson's equation

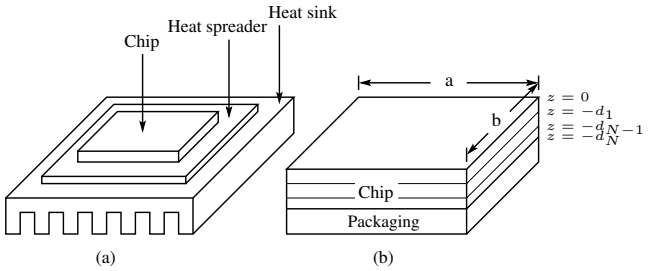


Fig. 1. Schematic of a VLSI chip with packaging (a) IC chip and the packaging structure (b) simplified model of the chip and packaging.

$$\nabla^2 T(\mathbf{r}) = -\frac{g(\mathbf{r})}{k_{l(\mathbf{r})}} \tag{1}$$

where $\mathbf{r} = (x, y, z)$, $T(\mathbf{r})$ is the temperature (°C) distribution inside the chip, $g(\mathbf{r})$ is the volume power density (W/m^3), and $k_{l(\mathbf{r})}$ is the thermal conductivity (W/(m·°C)) of the layer where point $\mathbf{r}$ is located [16]. The vertical surfaces and the top surface of the chip are assumed to be adiabatic [17], and the bottom surface of the chip is assumed to be convective, with an effective heat transfer coefficient h (W/(m^2·°C)) [18]. In mathematical form, these boundary conditions can be expressed as

$$\left.\frac{\partial T(\mathbf{r})}{\partial x}\right|_{x=0,a} = \left.\frac{\partial T(\mathbf{r})}{\partial y}\right|_{y=0,b} = 0 \tag{2}$$

$$\left.\frac{\partial T(\mathbf{r})}{\partial z}\right|_{z=0} = 0 \tag{3}$$

$$k_N \left.\frac{\partial T(\mathbf{r})}{\partial z}\right|_{z=-d_N} = h\left(T(\mathbf{r})|_{z=-d_N} - T_a\right) \tag{4}$$

where T_a is the ambient temperature, and k_N is the thermal conductivity of the bottom layer of the chip. In addition, we enforce the continuity conditions at the interface between adjacent layers within the multilayered chip, i.e.,

$$T(\mathbf{r})|_{z=-d_i+\epsilon} = T(\mathbf{r})|_{z=-d_i-\epsilon} \tag{5}$$

$$k_i \left.\frac{\partial T(\mathbf{r})}{\partial z}\right|_{z=-d_i+\epsilon} = k_{i+1} \left.\frac{\partial T(\mathbf{r})}{\partial z}\right|_{z=-d_i-\epsilon} \tag{6}$$

where ϵ is an infinitesimally small quantity and k_i is the thermal conductivity of the i^{th} material layer in the multilayered chip structure.

B. Green function for the rectangular-shaped multilayered structure

Let $G(\mathbf{r}, \mathbf{r}')$, with $\mathbf{r} = (x, y, z)$ and $\mathbf{r}' = (x', y', z')$, be the distribution of temperature above T_a in the multilayer when a unit point power source of 1W is placed at position $\mathbf{r}'$. Then $G(\mathbf{r}, \mathbf{r}')$ satisfies the equation

$$\nabla^2 G(\mathbf{r}, \mathbf{r}') = -\frac{\delta(\mathbf{r} - \mathbf{r}')}{k_{l(\mathbf{r})}} \tag{7}$$

and the boundary conditions

$$\left.\frac{\partial G(\mathbf{r},\mathbf{r}')}{\partial x}\right|_{x=0,a} = \left.\frac{\partial G(\mathbf{r},\mathbf{r}')}{\partial y}\right|_{y=0,b} = 0 \tag{8}$$

$$\left.\frac{\partial G(\mathbf{r},\mathbf{r}')}{\partial z}\right|_{z=0} = 0 \tag{9}$$

$$k_N \left.\frac{\partial G(\mathbf{r},\mathbf{r}')}{\partial z}\right|_{z=-d_N} = hG(\mathbf{r},\mathbf{r}')|_{z=-d_N} \tag{10}$$

$$G(\mathbf{r},\mathbf{r}')|_{z=-d_i+\epsilon} = G(\mathbf{r},\mathbf{r}')|_{z=-d_i-\epsilon} \tag{11}$$

$$k_i \left.\frac{\partial G(\mathbf{r},\mathbf{r}')}{\partial z}\right|_{z=-d_i+\epsilon} = k_{i+1} \left.\frac{\partial G(\mathbf{r},\mathbf{r}')}{\partial z}\right|_{z=-d_i-\epsilon} \tag{12}$$

where $\delta(\mathbf{r}-\mathbf{r}') = \delta(x-x')\delta(y-y')\delta(z-z')$ is the three-dimensional Dirac delta function, and $G(\mathbf{r},\mathbf{r}')$ is the Green function. The temperature field under an arbitrary power density distribution can be obtained easily as

$$T(\mathbf{r}) = T_a + \int_0^a dx' \int_0^b dy' \int_{-d_N}^0 dz' G(\mathbf{r},\mathbf{r}') g(\mathbf{r}') \tag{13}$$

As shown in [10] and [11] for electrical problems, the Green function can be generally written in the form

$$G(\mathbf{r},\mathbf{r}') = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \cos\left(\frac{m\pi x}{a}\right)\cos\left(\frac{n\pi y}{b}\right) \times \cos\left(\frac{m\pi x'}{a}\right)\cos\left(\frac{n\pi y'}{b}\right) Z'_{mn}(z,z') \tag{14}$$

where $Z'_{mn}(z,z')'s$ are functions of only the z coordinates of the source and field points. The specific form of each $Z'_{mn}(z,z')$ depends on the boundary conditions, and it can be derived similarly to that shown in [10] and [11].

In the following analysis, we assume that both the heat sources and the field regions are located on discrete horizontal planes. Since the vertical dimensions of the devices are much smaller than that of the silicon chip, this assumption is reasonable for most practical purposes. For a particular pair of source and field planes, i.e., for a particular z and z', the Green function can be written as

$$G(x,y,x',y') = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} C_{mn}\cos\left(\frac{m\pi x}{a}\right)\cos\left(\frac{n\pi y}{b}\right) \times \cos\left(\frac{m\pi x'}{a}\right)\cos\left(\frac{n\pi y'}{b}\right) \tag{15}$$

The temperature distribution on the field plane due to the heat sources on the source plane is given by

$$T(x,y) = T_a + \int_0^a dx' \int_0^b dy' G(x,y,x',y') P_d(x',y') \tag{16}$$

where $P_d(x',y')$ is the power density distribution on the source plane.

III. Thermal Simulation Algorithms

A. Algorithm I: Thermal simulation using the DCT and table look-up

Since practically all of the on-chip geometries can be decomposed into combinations of rectangles, we only focus on the rectangular-shaped source and field regions in the following analysis. Fig. 2 shows a schematic of a source and a field region. Note that the two regions can have different z coordinates if the field plane does not coincide with the source plane. Our objective here is to calculate the average temperature $\overline{T_f}$ of the field region efficiently given the power density P_d of the source region. To simplify the analysis, we assume that P_d is a constant within the source region. This is not a very restrictive assumption, since if the power density is not uniformly distributed in the source region, we can always divide the source region into smaller rectangular-shaped sub-regions and assume that the power density is uniform within each sub-region.

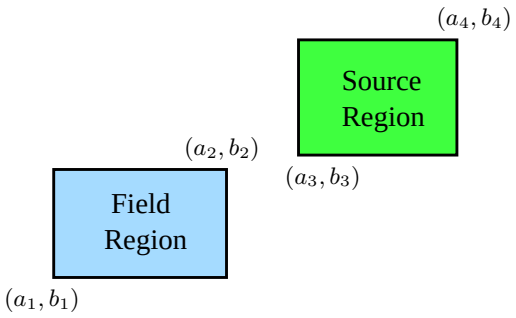


Fig. 2. Source and field regions for computing the temperature distribution.

The average temperature in the field region can be computed using

$$\overline{T_f} = \frac{1}{(a_2-a_1)(b_2-b_1)} \int_{a_1}^{a_2} dx \int_{b_1}^{b_2} dy T(x,y) \tag{17}$$

Substituting (15) and (16) into (17), and modifying the integration limits of (16) according to the location and dimensions of the source region, we obtain

$$\begin{aligned}
\overline{T_f} &= T_a + \frac{P_d}{(a_2-a_1)(b_2-b_1)} \times \\
&\int_{a_1}^{a_2} dx \int_{b_1}^{b_2} dy \int_{a_3}^{a_4} dx' \int_{b_3}^{b_4} dy' G(x,y,x',y') \\
&= T_a + C_{00} P_d (a_4-a_3)(b_4-b_3) + \\
&\left\{ \frac{P_d(b_4-b_3)}{(a_2-a_1)} \sum_{m=0}^{\infty} D_{m0} \left[\sin\left(\frac{m\pi a_2}{a}\right) - \sin\left(\frac{m\pi a_1}{a}\right)\right] \times \right. \\
&\left. \left[\sin\left(\frac{m\pi a_4}{a}\right) - \sin\left(\frac{m\pi a_3}{a}\right)\right] \right\} + \\
&\left\{ \frac{P_d(a_4-a_3)}{(b_2-b_1)} \sum_{n=0}^{\infty} E_{0n} \left[\sin\left(\frac{n\pi b_2}{b}\right) - \sin\left(\frac{n\pi b_1}{b}\right)\right] \times \right. \\
&\left. \left[\sin\left(\frac{n\pi b_4}{b}\right) - \sin\left(\frac{n\pi b_3}{b}\right)\right] \right\} + \\
&\left\{ \frac{P_d}{(a_2-a_1)(b_2-b_1)} \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} F_{mn} \left[\sin\left(\frac{m\pi a_2}{a}\right) - \sin\left(\frac{m\pi a_1}{a}\right)\right] \times \right. \\
&\left. \left[\sin\left(\frac{m\pi a_4}{a}\right) - \sin\left(\frac{m\pi a_3}{a}\right)\right] \left[\sin\left(\frac{n\pi b_2}{b}\right) - \sin\left(\frac{n\pi b_1}{b}\right)\right] \times \right. \\
&\left. \left[\sin\left(\frac{n\pi b_4}{b}\right) - \sin\left(\frac{n\pi b_3}{b}\right)\right] \right\}
\end{aligned} \tag{18}$$

where

$$D_{m0} = \begin{cases} C_{m0}\left(\frac{a}{m\pi}\right)^2 & \text{if } m{\neq}0 \\ 0 & \text{if } m = 0 \end{cases} \quad (19)$$

$$E_{0n} = \begin{cases} C_{0n}\left(\frac{b}{n\pi}\right)^2 & \text{if } n{\neq}0 \\ 0 & \text{if } n = 0 \end{cases} \quad (20)$$

$$F_{mn} = \begin{cases} C_{mn}\left(\frac{a}{m\pi}\right)^2\left(\frac{b}{n\pi}\right)^2 & \text{if } m{\neq}0, n{\neq}0 \\ 0 & \text{otherwise} \end{cases} \quad (21)$$

Using the identity

$$\sin(\theta_1)\sin(\theta_2) = \frac{1}{2}(\cos(\theta_1-\theta_2) - \cos(\theta_1+\theta_2)) \quad (22)$$

the first summation

$$\sum_{m=0}^{\infty} D_{m0}\left[\sin\left(\frac{m\pi a_2}{a}\right) - \sin\left(\frac{m\pi a_1}{a}\right)\right]\left[\sin\left(\frac{m\pi a_4}{a}\right) - \sin\left(\frac{m\pi a_3}{a}\right)\right] \quad (23)$$

can be re-written as a sum of eight terms in the form

$$\pm\frac{1}{2}\sum_{m=0}^{\infty} D_{m0}\cos\left(\frac{m\pi(a_i{\pm}a_j)}{a}\right) \quad (24)$$

where $i = 1, 2$ and $j = 3, 4$.

To utilize the DCT, we first discretize the source and field planes into M equal divisions along the x direction and N equal divisions along the y direction and form the grids. Then we truncate the summation in equation (24) at index M. As will be discussed later, the indices M and N are determined by the considerations of both the resolution of thermal analysis and the convergence of the Green function. If we assume that all the vertices of the field and source regions are located on grid points, i.e., $\frac{a_i}{a} = \frac{k_i}{M}$, $\frac{a_j}{a} = \frac{k_j}{M}$, where k_i and k_j are integers, and $0{\leq}k_i{\leq}M$, $0{\leq}k_j{\leq}M$, then equation (24) becomes

$$\pm\frac{1}{2}\sum_{m=0}^{M} D_{m0}\cos\left(\frac{m\pi(k_i{\pm}k_j)}{M}\right) \quad (25)$$

Let

$$k = \begin{cases} k_i{\pm}k_j & \text{if } 0{\leq}k_i{\pm}k_j{\leq}M \\ -(k_i{\pm}k_j) & \text{if } k_i{\pm}k_j < 0 \\ 2M-(k_i{\pm}k_j) & \text{if } k_i{\pm}k_j > M \end{cases} \quad (26)$$

then $0{\leq}k{\leq}M$ and equation (25) can be re-written as

$$\pm\frac{1}{2}\sum_{m=0}^{M} D_{m0}\cos\left(\frac{m\pi k}{M}\right) \quad (27)$$

This is precisely one term in the type-I DCT of the sequence D_{m0}, and the DCT sequence can be computed efficiently using the fast Fourier transform (FFT) in $O(M\log(M))$ time [19]. After the DCT sequence is obtained, it can be stored in a vector and used many times in future temperature calculations. As a result, the computation of summation (23) is reduced to eight look-ups in the DCT vector in constant time and then adding up the look-up results. Similarly, the summation involving E_{0n} in equation (18) can also be obtained efficiently using the DCT and table look-ups.

The double summation in equation (18) can be re-written as a sum of 64 terms in the form

$$\pm\frac{1}{4}\sum_{m=0}^{\infty}\sum_{n=0}^{\infty} F_{mn}\cos\left(\frac{m\pi(a_i{\pm}a_j)}{a}\right)\cos\left(\frac{n\pi(b_p{\pm}b_q)}{b}\right) \quad (28)$$

where $i = 1, 2$, $j = 3, 4$, $p = 1, 2$, and $q = 3, 4$. Using a similar approach, equation (28) can be cast into

$$\pm\frac{1}{4}\sum_{m=0}^{M}\sum_{n=0}^{N} F_{mn}\cos\left(\frac{m\pi k}{M}\right)\cos\left(\frac{n\pi l}{N}\right) \quad (29)$$

where $0{\leq}k{\leq}M$ and $0{\leq}l{\leq}N$. This is one term in the 2-D type-I DCT of the matrix F_{mn}. The 2-D DCT matrix can be computed using the FFT in $O((M{\cdot}N)\times\log(M{\cdot}N))$ time, and after the 2-D DCT table is obtained, the double summation reduces to 64 table look-ups in constant time and then adding up the look-up results.

Note that when multiple heat sources are present, their effects on the average temperature rise above T_a in the field region, i.e., the integral term in equation (16), can be summed up to obtain the total average temperature rise.

The selection of the discretization parameters M and N deserves some more considerations. Assume that the minimum feature size along the x and y directions that must be resolved are x_{min} and y_{min}, respectively, then M and N must satisfy

$$M{\geq}M_r = \frac{a}{x_{min}} \quad \text{and} \quad N{\geq}N_r = \frac{b}{y_{min}} \quad (30)$$

where M_r and N_r represent the minimum values of M and N from resolution considerations. However, since M and N are also the truncation points of the summations in equation (18), they must be large enough to ensure the convergence of the summations. As pointed out in [20], the summations converge more slowly as x_{min} and y_{min} become smaller relative to the chip dimensions a and b. Thus, the actual values of M and N cannot be determined merely based on M_r and N_r. Let M_c and N_c be the minimum values of $M{\geq}M_r$ and $N{\geq}N_r$ such that the convergence is achieved in (18). In our implementation, M_c and N_c are determined as follows. We consider nine representative regions on each of the source and field planes as shown in Fig. 3. Each region has dimensions of $x_{min}{\times}y_{min}$. We first set $M_c = M_r$ and $N_c = N_r$. Then we increase M_c and N_c gradually until the convergence of the summations in (18) is achieved for all of the possible locations of the source and field regions provided the source region coincides with one of the nine representative regions on the source plane while the field region coincides with one of the nine representative regions on the field plane. Finally, to assist the utilization of the FFT in the DCT computations, M and N are chosen to be integers that are powers of 2 and are no smaller than M_c and N_c, respectively.

Compared with the classical Green function method, the advantage of our algorithm is that it replaces the expensive double summations in the expressions involving the Green function by the inexpensive summations of a few numbers in the pre-calculated look-up table and vectors. The look-up table and vectors only depend on the chip dimensions and the physical properties of the substrate, but are independent of the layout and power distribution. Hence, the look-up table and vectors

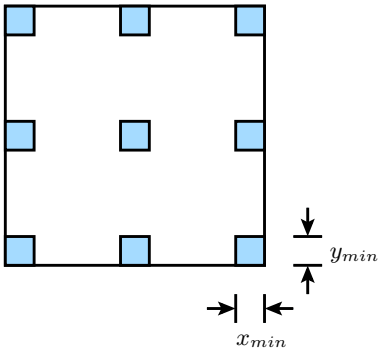


Fig. 3. The locations of the nine representative regions on the source plane. Each region has dimensions of $x_{min} \times y_{min}$. One region is located at the center of the plane, one is at the mid-point of each edge, and one is at each corner. Similarly, we have nine representative regions on the field plane.

can be calculated once and then used many times in thermal-aware physical designs, which significantly reduces the amortized cost of obtaining the table and vectors, and improves the overall efficiency of the algorithm.

B. Algorithm II: Full-chip thermal simulation using the spectral domain computations

Algorithm I gained its efficiency from the faster evaluations of the expressions involving the Green function. Asymptotically, however, it is still an expensive method for simulations involving a large number of heat sources and field regions because the effects of the heat sources on the field regions are calculated in a pair-wise fashion. The second algorithm we present in this section targets full-chip thermal simulations with large problem sizes. It uses spectral domain analysis to reduce the asymptotic time complexity of calculating the on-chip temperature distribution. In the following analysis, we focus on the effect of one source plane on the temperature distribution in the field plane. When multiple source planes are present, their effects can be easily summed up to obtain the final solution.

Since the convolution integral in (16) can be considered as the governing equation of a linear system determined by the Green function $G(x, y, x', y')$, we can use spectral domain analysis to accelerate the computations corresponding to the convolution integral.

The first step of our algorithm is to obtain the spectral domain representation of the power density map in the form

$$P_d(x', y') = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} a_{ij} \phi_{ij}(x', y') \tag{31}$$

where

$$\phi_{ij}(x, y) = \cos\left(\frac{i\pi x}{a}\right) \cos\left(\frac{j\pi y}{b}\right) \tag{32}$$

It is easy to show that $\phi_{ij}(x, y)$ satisfies the equation

$$\lambda_{ij} \phi_{ij}(x, y) = \int_0^a dx' \int_0^b dy' G(x, y, x', y') \phi_{ij}(x', y') \tag{33}$$

where

$$\lambda_{ij} = \begin{cases} abC_{ij} & \text{if } i = j = 0 \\ \frac{1}{2} abC_{ij} & \text{if } i = 0, j \neq 0 \text{ or } i \neq 0, j = 0 \\ \frac{1}{4} abC_{ij} & \text{if } i \neq 0, j \neq 0 \end{cases} \tag{34}$$

is the response of the linear system to the spectral component $\phi_{ij}(x, y)$ [12]. After the spectral domain representation of the power density distribution in the source plane is obtained, the temperature distribution in the field plane can be calculated easily by

$$T(x, y) = T_a + \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} \lambda_{ij} a_{ij} \phi_{ij}(x, y) \tag{35}$$

As will be shown next, both the spectral decomposition in (31) and the double-summation in (35) can be calculated efficiently using the DCT and IDCT through the FFT.

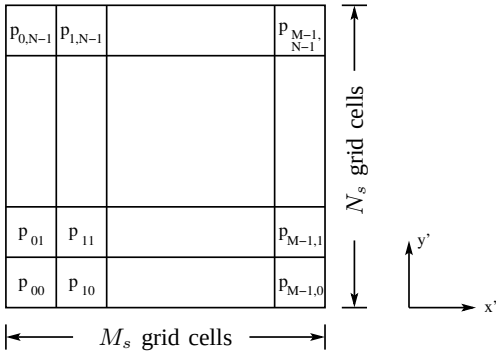


Fig. 4. The arrangement of the $M_s \times N_s$ grid cells on the source plane.

Now we assume that the source plane is divided into $M_s \times N_s$ rectangular grid cells of equal size as shown in Fig. 4, and the power density in each grid cell on the source plane is uniform, i.e., the power density distribution can be written in the piece-wise constant form

$$P_d(x', y') = \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn} \Theta(x' - (m+\frac{1}{2})\Delta x_s, y' - (n+\frac{1}{2})\Delta y_s) \tag{36}$$

where

$$\Theta(x', y') = \begin{cases} 1 & \text{if } |x'| \leq \frac{1}{2}\Delta x_s \text{ and } |y'| \leq \frac{1}{2}\Delta y_s \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

and $\Delta x_s = \frac{a}{M_s}, \Delta y_s = \frac{b}{N_s}$. P_{mn} is the power density of the mn^{th} grid cell.

Note that if the piece-wise constant power density map is not directly given in the form of (36), it can be conveniently derived from the layout geometries and the power generated by each circuit component. Assume that the layout of each component is within a rectangular-shaped region as shown in Fig. 5, and the region corresponding to the i^{th} component C_i is defined by $x_i^L \leq x \leq x_i^R$ and $y_i^B \leq y \leq y_i^T$. The range of the indices m and n of the grid cells that the i^{th} component overlaps is given

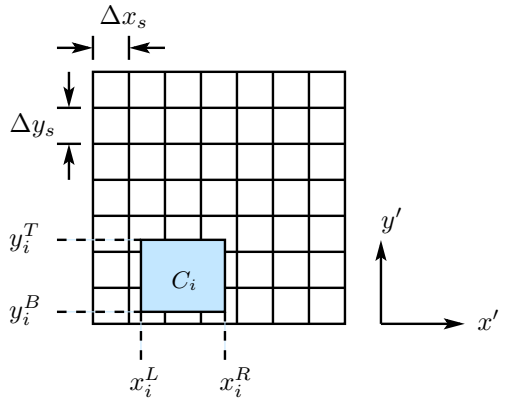


Fig. 5. Calculating the power density map from the given layout geometries and the power generated by each circuit component.

by

$$\begin{aligned} \lfloor \frac{x_i^L}{\Delta x_s} \rfloor \leq m \leq \lfloor \frac{x_i^R}{\Delta x_s} \rfloor \\ \lfloor \frac{y_i^B}{\Delta y_s} \rfloor \leq n \leq \lfloor \frac{y_i^T}{\Delta y_s} \rfloor \end{aligned} \tag{38}$$

Assume that the total power generated by the i^{th} component is given by P_i^T, then its contribution to the power density of the mn^{th} grid cell that overlaps with it is

$$\delta P_{mn}^i = P_i^T \times \frac{S_{mn}^i}{(x_i^R - x_i^L)(y_i^T - y_i^B)} \times \frac{1}{\Delta x_s \cdot \Delta y_s} \tag{39}$$

where S_{mn}^i is the overlap area of the rectangle corresponding to the i^{th} component and the mn^{th} rectangular-shaped grid cell, and it can be calculated in constant time. Therefore, obtaining the piece-wise constant power density map from the layout geometries and the power generated by each circuit component has only a linear time complexity with respect to the number of components in the circuit, and it can be usually ignored compared with the costs of other calculations involved in the thermal simulation.

Substituting (36) into (31) and using the orthogonality property of the cosine functions in the integral sense, we obtain

$$a_{ij} = A_{ij} \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn} \cos\left(\frac{i\pi(2m+1)}{2M_s}\right) \cos\left(\frac{j\pi(2n+1)}{2N_s}\right) \tag{40}$$

where

$$A_{ij} = \begin{cases} \frac{1}{M_s N_s} & \text{if } i = j = 0 \\ \frac{4}{iN_s\pi} \sin\left(\frac{i\pi}{2M_s}\right) & \text{if } i \neq 0, j = 0 \\ \frac{4}{M_s j\pi} \sin\left(\frac{j\pi}{2N_s}\right) & \text{if } i = 0, j \neq 0 \\ \frac{16}{ij\pi^2} \sin\left(\frac{i\pi}{2M_s}\right) \sin\left(\frac{j\pi}{2N_s}\right) & \text{if } i \neq 0, j \neq 0 \end{cases} \tag{41}$$

Note that to accurately represent the power density distribution $P_d(x', y')$ using (31), the theoretical upper limit of the double summation should be infinity. In practical implementations, however, the summation must be truncated to ensure a reasonable runtime. Since (31) is essentially the Fourier expansion of $P_d(x', y')$, a natural criterion for determining the truncation point is that enough "energy" contained in $P_d(x', y')$ is covered by the truncated Fourier expansion. Mathematically, we have

$$\int_0^a dx' \int_0^b dy' P_d^2(x', y') = ab \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} s_{ij} a_{ij}^2 \tag{42}$$

where

$$s_{ij} = \begin{cases} 1 & \text{if } i = j = 0 \\ \frac{1}{2} & \text{if } i = 0, j \neq 0 \text{ or } i \neq 0, j = 0 \\ \frac{1}{4} & \text{if } i \neq 0, j \neq 0 \end{cases} \tag{43}$$

Substituting (36) into the left hand side of (42), we obtain

$$\frac{1}{M_s N_s} \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn}^2 = \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} s_{ij} a_{ij}^2 \tag{44}$$

which can be considered as a form of the Parseval's theorem. The truncation points M' and N' are then determined by

$$\sum_{i=0}^{M'-1} \sum_{j=0}^{N'-1} s_{ij} a_{ij}^2 \geq \eta \left(\frac{1}{M_s N_s} \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn}^2 \right) \tag{45}$$

where η is the proportion of the "energy" of the space domain signal $P_d(x', y')$ that must be covered by the truncated Fourier expansion. In practice, we found that setting η to 90% will usually be enough to obtain very accurate results in temperature calculations.

We emphasize here that (45) does not imply that only a fraction, η, of the total power generated by the heat sources is included in the truncated expansion. In reality, the total power is completely contained in the DC term of expansion (31), and (45) only describes how accurately we are approximating the *exact* shape of the space domain signal, i.e., $P_d(x', y')$. A smaller η implies that more components with high spectral numbers in $P_d(x', y')$ are ignored, or equivalently, more *zero mean* noises with high spectral numbers are added to the approximating power distribution. Since the temperature distribution is calculated using (16) and the convolution with the Green function has a low-pass filtering effect, η does not have to be extremely close to 1 in order to calculate the temperature accurately. We also point out that although η is set to a constant number, the truncation points M' and N' are not determined *a priori* in our algorithm. Instead, they depend on $P_d(x', y')$ according to (45). Our strategy of determining the truncation points is to first set $M' = M_s$ and $N' = N_s$. If (45) is not satisfied, then we increase M' to $2M_s$ and N' to $2N_s$. The summation limits M' and N' continue to increase with steps of M_s and N_s until (45) is satisfied. The importance of determining the truncation points dynamically based on the input data will become more obvious as the size of the problem increases.

Note that for $0 \leq i < M_s$ and $0 \leq j < N_s$, the double summation in (40) can be considered as a term in the 2D type-II DCT [19] of the power density matrix P. For $i \geq M_s$ or $j \geq N_s$, we can always find integers s_1 and s_2 such that $i = 2s_1 M_s \pm \hat{i}$ and $j =$

$2s_2N_s \pm \hat{j}$ where $0 \leq \hat{i} < M_s$ and $0 \leq \hat{j} < N_s$[1]. Hence, for any i and j, we always have

$$a_{ij} = \pm A_{ij} \widetilde{P}_{\hat{i}\hat{j}} \tag{46}$$

where

$$\widetilde{P}_{\hat{i}\hat{j}} = \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn} \cos\left(\frac{\hat{i}\pi(2m+1)}{2M_s}\right) \cos\left(\frac{\hat{j}\pi(2n+1)}{2N_s}\right) \tag{47}$$

with $0 \leq \hat{i} < M_s$ and $0 \leq \hat{j} < N_s$ is the 2D type-II DCT of the P matrix and the sign of (46) is determined by whether s_1 and s_2 are even or odd numbers [12]. Equation (47) can be calculated efficiently using the 2D FFT in $O((M_s \cdot N_s) \times \log(M_s \cdot N_s))$ time. After the 2D DCT matrix $\widetilde{P}$ is obtained, the calculation of a_{ij} simply involves computing the coefficient A_{ij} and finding the corresponding term $\widetilde{P}_{\hat{i}\hat{j}}$.

From (32), (35), and (45), the temperature distribution $T(x, y)$ can now be written as

$$T(x,y) = T_a + \sum_{i=0}^{M'-1} \sum_{j=0}^{N'-1} \lambda_{ij} a_{ij} \cos\left(\frac{i\pi x}{a}\right) \cos\left(\frac{j\pi y}{b}\right) \tag{48}$$

If we assume that the temperature field plane is divided into $M_f \times N_f$ rectangular grid cells of equal size, then the average temperature of the mn^{th} grid cell can be obtained by

$$\begin{aligned} T_{mn} &= \frac{1}{\Delta x_f \Delta y_f} \int_{m\Delta x_f}^{(m+1)\Delta x_f} dx \int_{n\Delta y_f}^{(n+1)\Delta y_f} dy T(x,y) \\ &= T_a + \sum_{i=0}^{M'-1} \sum_{j=0}^{N'-1} B_{ij} \cos\left(\frac{i\pi(2m+1)}{2M_f}\right) \cos\left(\frac{j\pi(2n+1)}{2N_f}\right) \end{aligned} \tag{49}$$

where $\Delta x_f = \frac{a}{M_f}$, $\Delta y_f = \frac{b}{N_f}$, and

$$B_{ij} = \begin{cases} \lambda_{ij} a_{ij} & \text{if } i = j = 0 \\ 2\lambda_{ij} a_{ij} \frac{M_f}{i\pi} \sin\left(\frac{i\pi}{2M_f}\right) & \text{if } i \neq 0, j = 0 \\ 2\lambda_{ij} a_{ij} \frac{N_f}{j\pi} \sin\left(\frac{j\pi}{2N_f}\right) & \text{if } i = 0, j \neq 0 \\ 4\lambda_{ij} a_{ij} \frac{M_f N_f}{ij\pi^2} \sin\left(\frac{i\pi}{2M_f}\right) \sin\left(\frac{j\pi}{2N_f}\right) & \text{if } i \neq 0, j \neq 0 \end{cases} \tag{50}$$

Similar to the analysis shown previously, any $i \geq M_f$ and $j \geq N_f$ can be written as $i = 2s_3 M_f \pm \hat{i}$ and $j = 2s_4 N_f \pm \hat{j}$ such that $0 \leq \hat{i} < M_f$, $0 \leq \hat{j} < N_f$, and s_3 and s_4 are integers. Using the periodicity of the cosine function, we can finally cast T_{mn} into the form

$$T_{mn} = T_a + \sum_{\hat{i}=0}^{M_f-1} \sum_{\hat{j}=0}^{N_f-1} L_{\hat{i}\hat{j}} \cos\left(\frac{\hat{i}\pi(2m+1)}{2M_f}\right) \cos\left(\frac{\hat{j}\pi(2n+1)}{2N_f}\right) \tag{51}$$

[1]If i equals an odd multiple of M_s, we will not be able to write i as $i = 2s_1 M_s \pm \hat{i}$. However, for this kind of i, it can be easily shown that $a_{ij} = 0$ because $\cos\left(\frac{i\pi(2m+1)}{2M_s}\right) = 0$. Similarly, we know that $a_{ij} = 0$ if j equals an odd multiple of N_s.

Input:
- Chip geometry and physical properties of the material layers.
- Power density map - matrix P.

Output: Temperature distribution map - matrix T.
Algorithm:
1) Calculate the Green function coefficients $C_{ij}{}'s$;
2) Calculate the spectral responses of the system $\lambda_{ij}{}'s$;
3) Calculate the type-II 2D DCT of the power density matrix $\widetilde{P} = \text{2DDCT}(P)$;
4) $TSE = \frac{1}{M_s N_s} \sum_{m=0}^{M_s-1} \sum_{n=0}^{N_s-1} P_{mn}^2$;
5) $M' = M_s$, $N' = N_s$;
   $ASE = \sum_{i=0}^{M'-1} \sum_{j=0}^{N'-1} s_{ij} a_{ij}^2$;
   while ($ASE < \eta \times TSE$)
      $M' = M' + M_s$, $N' = N' + N_s$;
      Update ASE;
   end while;
6) Calculate the matrix L;
7) Calculate the temperature distribution map using the type-II 2D IDCT $T = T_a + \text{2DIDCT}(L)$;

Fig. 6. Thermal simulation algorithm using the Green function method, the DCT, and the spectral domain computations.

where

$$L_{\hat{i}\hat{j}} = \begin{cases} B_{00} & \text{if } \hat{i} = \hat{j} = 0 \\ \sum_{\substack{i < M' \\ i = 2s_3 M_f \pm \hat{i}}} \pm B_{i0} & \text{if } \hat{i} \neq 0, \hat{j} = 0 \\ \sum_{\substack{j < N' \\ j = 2s_4 N_f \pm \hat{j}}} \pm B_{0j} & \text{if } \hat{i} = 0, \hat{j} \neq 0 \\ \sum_{\substack{i < M' \\ i = 2s_3 M_f \pm \hat{i}}} \sum_{\substack{j < N' \\ j = 2s_4 N_f \pm \hat{j}}} \pm B_{ij} & \text{if } \hat{i} \neq 0, \hat{j} \neq 0 \end{cases} \tag{52}$$

and the signs of the $B's$ in (52) are determined by whether s_3 and s_4 are even or odd numbers. After the matrix L is obtained, the double summation in (51) can be calculated efficiently using the 2D IDCT.

The complete thermal simulation algorithm using the Green function method, the DCT, and the spectral domain computations is shown in Fig. 6. The asymptotic time complexity of the algorithm is $O(\mathcal{N}_{gs} \times \log(\mathcal{N}_{gs})) + O(\mathcal{N}_{gf} \times \log(\mathcal{N}_{gf}))$ where $\mathcal{N}_{gs} = M_s \cdot N_s$ is the total number of grid cells in the power density map, and $\mathcal{N}_{gf} = M_f \cdot N_f$ is the total number of grid cells in the resulting temperature profile. This is a significant improvement over the $O(\mathcal{N}_{gs} \cdot \mathcal{N}_{gf})$ complexity of Algorithm I for full-chip thermal simulations.

C. Algorithm III: Thermal simulation with local high accuracy requirements

Although Algorithm II can achieve a $O(\mathcal{N}_{gs} \times \log(\mathcal{N}_{gs})) + O(\mathcal{N}_{gf} \times \log(\mathcal{N}_{gf}))$ time complexity as opposed to a $O(\mathcal{N}_{gs} \cdot \mathcal{N}_{gf})$ complexity of Algorithm I for full-chip thermal simulations, Algorithm I is still more efficient for performing the localized analysis, where the effects of a few critical circuit

blocks on the temperature distribution in a few key field regions are of interest. This is because to apply Algorithm II, we must always superimpose regular grids over the entire source and field planes and calculate the complete temperature profile from the complete power density distribution. The size of each grid cell must be comparable with that of the resolution requirement of the calculation, and the total number of grid cells determines the problem size. Therefore, although Algorithm II has a smaller asymptotic time complexity than Algorithm I for full-chip thermal simulations, it may also require the formulation of a problem with much larger size than Algorithm I if only some localized temperature calculations are required by circuit designers.

We will face an even more difficult decision concerning whether Algorithm I or Algorithm II should be used when a circuit designer has different requirements on the accuracy of the thermal simulation over different parts of the same chip. For example, in mixed signal designs where analog circuits are fabricated on the same chip as digital circuits, the analog blocks often have more stringent accuracy requirements on the thermal simulation because the operations of the analog circuits are more sensitive to temperature. If the full-chip temperature profile is required, then Algorithm I will be too slow to use. However, in order to use Algorithm II, the size of each grid cell must be small enough so that the high accuracy requirements of the analog blocks are satisfied. This may result in very dense grids and a large problem size. For these kinds of problems, a better strategy can be adopted to accelerate the runtime of the algorithm further by combining the advantages of both Algorithm I and II. The key idea is to use coarse grids to divide the source and field planes where each grid cell in the source plane can contain several logic gates or analog functional units, and the size of each grid cell in the field plane satisfies the accuracy requirements of the digital circuits. The power density of each grid cell in the source plane is calculated by summing up the power dissipations of all the logic gates and analog functional units located in it and dividing the sum by the area of the grid cell. A coarse temperature map for the field plane is then obtained from the coarse power density map using Algorithm II and is used for the digital blocks. Finally, for each analog functional unit on the field plane whose temperature is to be calculated more accurately, we use Algorithm I to compute the contributions to its temperature rise from the nearby logic gates and analog function units on the source plane, and use this result to correct the temperature obtained by Algorithm II over the coarse grid cell. To simplify the presentation, we assume in the following analysis that the source plane coincides with the field plane and both of them are divided into $M \times N$ coarse grid cells. However, this assumption is not essential to the algorithm and it can be relaxed easily to handle multiple source and field planes such as that in the emerging three-dimensional IC technologies.

Fig. 7 shows a chip that is divided into $M \times N$ coarse grid cells each of which contains several logic gates or analog functional units, and let the shaded area represent the analog block. An $M \times N$ temperature map is first obtained. The inaccuracies in the temperature calculations, besides that due to the truncation

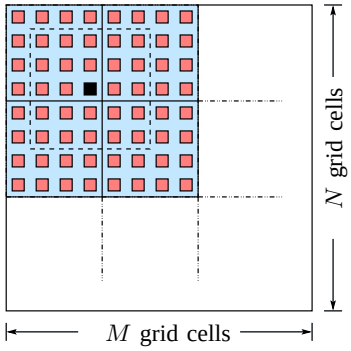


Fig. 7. A mixed signal chip where the analog block has higher requirement on the accuracy of thermal simulations. The logic gates and analog functional units within the dashed line constitute the set $C(A)$.

of the spectral domain representation of the power density map, will come from two sources which include

- Assuming that the power density in each grid cell is uniform.
- Only the average temperature of each grid cell is calculated, i.e., all of the logic gates and analog functional units inside the same grid cell obtain the same calculated temperature.

Now assume that we need to calculate the temperature of the analog functional unit A located in the ij^{th} grid cell and represented by the black rectangle more accurately. Let T_{ij} be the average temperature of the ij^{th} grid cell obtained using Algorithm II, and let $T_{ij,S}$ be the contribution to the average temperature rise of the ij^{th} grid cell from the logic gate or analog functional unit S assuming that the power generated by S is uniformly distributed in the grid cell in which it resides. Denote the more accurate average temperature of the analog functional unit A by $T_A^{accurate}$, and let $T_{A,S}^{accurate}$ be the accurate contribution to the temperature rise of A from the logic gate or analog functional unit S. The temperature $T_A^{accurate}$ can be obtained by

$$T_A^{accurate} = T_{ij} - \sum_{S \in C(A)} T_{ij,S} + \sum_{S \in C(A)} T_{A,S}^{accurate} \qquad (53)$$

where $C(A)$, which will be called the interaction set of A in the following analysis, is the set of logic gates and analog functional units that are physically close to A, and hence, whose contributions to the temperature rise of A must be re-calculated accurately. The size of $C(A)$ is determined by the actual accuracy requirement on the temperature of A, and a higher accuracy requirement is usually associated with a larger $C(A)$. Both $T_{ij,S}$ and $T_{A,S}^{accurate}$ can be calculated efficiently using Algorithm I, and the overall efficiency of the combined algorithm is higher than that of Algorithm II applied with a fine grid over the entire chip that satisfies the high accuracy requirements of the analog functional units.

D. Time complexity analysis

We summarize the time complexities of the three algorithms in this section. Note that the calculations involved in each of the algorithms can be divided into two parts, i.e., those only

depend on the chip geometry and the physical properties of the chip materials, and those depend on the input power density distribution. The computation steps that only involve the chip geometry and material properties can be performed in the pre-characterization phase of the design, and their results can be stored for further uses. Therefore, the amortized costs of these steps are usually rather low in the overall physical design process, where the optimization routine executes the thermal simulation many times. The steps that involve the input power density distribution, however, must be executed within the optimization routine in physical designs. Hence, they usually dominate the overall runtime of the thermal-aware physical design algorithms such as the thermal-aware floorplanning and placement. The establishment of the look-up table and vectors in Algorithm I and the calculation of the spectral responses of the linear system in Algorithm II can both be performed in the pre-characterization phase, and in the following analysis, we will ignore the costs of these steps and only focus on the time complexity of the calculations that depend on the input power density distribution.

For the input-power-dependent steps in thermal simulations, Algorithm I has a time complexity of $O(\mathcal{N}_s \times \mathcal{N}_f)$, where $\mathcal{N}_s$ and $\mathcal{N}_f$ are the number of heat sources and field regions, respectively. Algorithm II always works with full-chip power density distribution and generates the complete on-chip temperature profile. It has a time complexity of $O(\mathcal{N}_{gs} \times \log(\mathcal{N}_{gs})) + O(\mathcal{N}_{gf} \times \log(\mathcal{N}_{gf}))$ where $\mathcal{N}_{gs} = M_s \cdot N_s$ is the total number of grid cells in the input power density map, and $\mathcal{N}_{gf} = M_f \cdot N_f$ is the total number of grid cells in the obtained temperature profile. Here, M_s and N_s are the number of grid divisions along the x and y directions on the source plane, and M_f and N_f are the number of grid divisions along the x and y directions on the field plane. It is obvious that Algorithm II is better than Algorithm I for full-chip temperature profiling, because the latter has a time complexity of $O(\mathcal{N}_{gs} \cdot \mathcal{N}_{gf})$. For the localized analysis where only a few source and field regions are involved, however, Algorithm I can often perform better because $\mathcal{N}_{gs}$ and $\mathcal{N}_{gf}$ are determined by the highest resolution requirement of the analysis, and $\mathcal{N}_s$ and $\mathcal{N}_f$ are usually much smaller than $\mathcal{N}_{gs}$ and $\mathcal{N}_{gf}$ for this type of problems.

To compare Algorithm II and Algorithm III, we assume that there are $\mathcal{N}_{total}$ logic gates and analog functional units in the design. Using Algorithm II directly with a grid size comparable to the smallest size of the gates and functional units will result in a time complexity of $O(\mathcal{N}_{total} \times \log(\mathcal{N}_{total}))$. For Algorithm III, a coarse grid is first used in the calculation. If we assume that each coarse grid cell contains K gates and functional units, then the time it takes to obtain the coarse temperature profile is $O(\frac{\mathcal{N}_{total}}{K} \times \log(\frac{\mathcal{N}_{total}}{K}))$. Now, if the accurate temperature correction is to be performed over all of the gates and functional units, then an additional cost of $O(\mathcal{N}_{total} \cdot K')$ is required where K' is the size of the interaction set of each gate or functional unit, and the total cost becomes $O(\mathcal{N}_{total} \times (\frac{1}{K}\log(\frac{\mathcal{N}_{total}}{K}) + K'))$. Note that the $O(\mathcal{N}_{total} \cdot K')$ term in the complexity analysis involves a relatively large pre-factor due to the 80 look-ups needed to calculate the correction corresponding to a pair of gates or functional units. Hence, the actual runtime of Algorithm III is often longer than that of Algorithm II when the accurate temperature correction is to be performed over all of the gates and functional units. However, as pointed out previously, it frequently happens in real design environments that the temperature correction is only required for a small portion of the circuit. Therefore, the total cost of Algorithm III becomes $O(\frac{\mathcal{N}_{total}}{K} \times \log(\frac{\mathcal{N}_{total}}{K}) + \mathcal{N}_c \cdot K')$, where $\mathcal{N}_c$ is the number of gates and functional units that require temperature corrections, and Algorithm III becomes more efficient than Algorithm II under this situation.

IV. Experimental Results

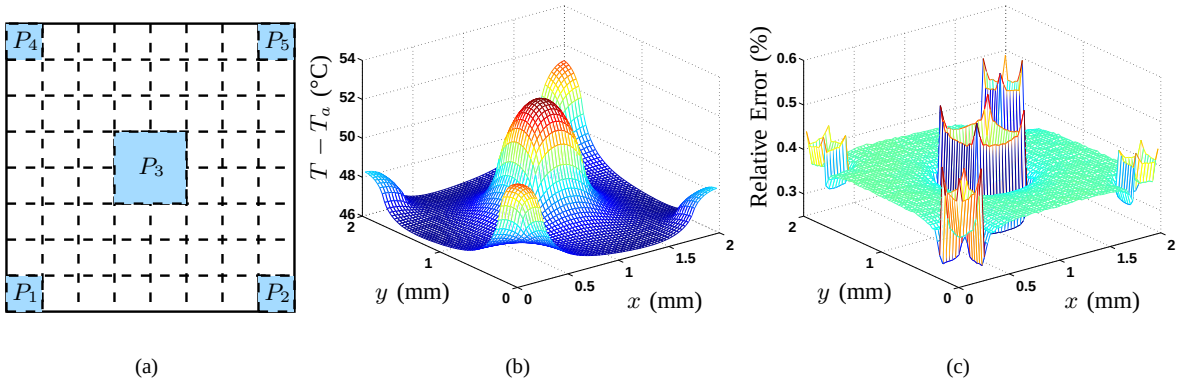
In this section, we present in detail the performance of the three algorithms, which are implemented in C++ and compiled using the level 3 optimization of g++. The experiments are performed on a desktop with a 3.2GHz Intel Pentium-4 CPU running the Red Hat Linux 8.0 operating system. We first compare the results obtained from Algorithm I with that from a commercial computational fluid dynamic (CFD) software package and that from the direct application of the Green function method in terms of accuracy and efficiency. Then we use Algorithm I as our base method to characterize the performance of the other two algorithms.

The commercial CFD software package uses a finite volume approach which meshes the entire substrate. Because of the discretized nature of the method, meshing errors are unavoidable. In order to control the meshing errors while still complete the computation within a reasonable amount of time, we start with a relatively rough mesh and continue refining it and re-running the simulation until the maximum error converges to around 1%. By doing this, we ensure that the result produced by the CFD software itself is accurate, and therefore it can be used as a valid criterion to evaluate the accuracy of our algorithms.

A. Accuracy and efficiency of Algorithm I

Fig. 8(a) shows the top surface of a silicon chip with dimensions of $2\text{mm} \times 2\text{mm} \times 0.5\text{mm}$. The area is divided into 8×8 equal square sections, and five power sources are placed in the corresponding sections as shown in the figure. The thermal conductivity k of silicon is $148\text{W/(m}\cdot\text{°C)}$, and the effective heat transfer coefficient h of the bottom surface of the chip is chosen to be $8700\text{W/(m}^2\cdot\text{°C)}$, which is consistent with the value used in [18]. The strength of the five power sources are $(P_1, P_2, P_3, P_4, P_5)$ = (0.2W, 0.1W, 1W, 0.1W, 0.2W).

Fig. 8(b) shows the top surface temperature map obtained using Algorithm I, where $T - T_a$ is the temperature rise above the ambient. In obtaining the temperature map, the top surface of the chip was divided into 64×64 small square regions with equal size and the average temperature in each small square region was computed. The parameters M and N were both set to 64, the minimum required values from resolution considerations, because the convergence of the Green function has already been achieved with $M = N = 64$. Fig. 8(c) shows the relative error in the temperature map compared with the computation result obtained from a commercial CFD software package for thermal analysis. We can see clearly that the error is below 1%, which demonstrates the accuracy of our method.



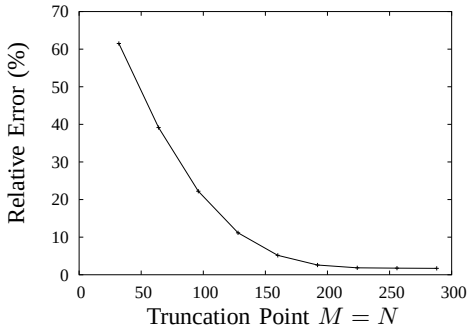
(a) (b) (c)

Fig. 8. Accuracy of Algorithm I (a) power source locations (b) computed temperature distribution above T_a using the proposed algorithm (c) relative error of the proposed algorithm compared with the result from a commercial CFD software package.

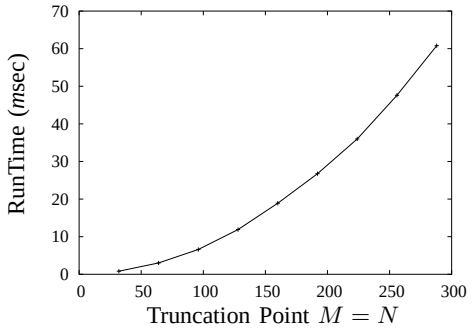
We next compare the efficiency of Algorithm I with that of the direct application of the Green function method to compute the temperature distribution. We still use the same chip dimensions and physical properties as in the previous example. However, only one power source is used this time to make the presentation clearer. The power source occupies a square region with dimensions of $\frac{2}{128}\text{mm} \times \frac{2}{128}\text{mm}$ at the exact center of the chip. The strength of the power source is $P_s = 50\text{mW}$. The average temperature above T_a of the source region itself is computed. The parameters M and N are both chosen to be 512 in our algorithm from convergence considerations for the Green function, i.e., we require the truncation error to be within 1%. The infinite summations in the Green function are more difficult to converge in this example because the sizes of the source and field regions relative to the chip dimensions are smaller than those in the previous example.

Using Algorithm I, the average temperature of the source region itself above T_a is found to be 11.537°C. The total computation time using the pre-calculated look-up table and vectors is only 5.5×10^{-4}*m*sec. As a comparison, we also computed the average temperature above T_a of the source region using equation (18) directly, which corresponds to the direct application of the Green function method. In the direct method, it is unnecessary to consider the resolution issue because equation (18) does not require the vertices (a_i, b_i) of the source and field regions to coincide with some grid points. So the parameters M and N are completely determined by the convergence consideration. Since the chip is square, we set $M = N$ in our analysis.

Fig. 9 shows the relative error and the corresponding runtime of the direct method. We can observe from the figure that even for a 5% relative error in $T - T_a$, the truncation point must be higher than 160. The runtime at this truncation point is 19*m*sec, which is four orders of magnitude slower than our algorithm, and the accuracy of our algorithm is much higher.



(a)



(b)

Fig. 9. Accuracy and computation time of the direct application of the Green function method (a) relative error in $T - T_a$ versus truncation point (b) runtime versus truncation point.

B. Comparison between Algorithm I and Algorithm II

Now, we compare the efficiency and accuracy of Algorithm I and II using a real chip example. Fig 10(a) shows a floorplan from [21], which is similar to that of the DEC Alpha 21264 processor but is scaled from the 350nm to the 65nm technology. The scaled chip dimensions are $3.3\text{mm} \times 3.3\text{mm} \times 0.506\text{mm}$, and we assume that the chip has the same physical properties as those used in the previous examples except that a layer representing the

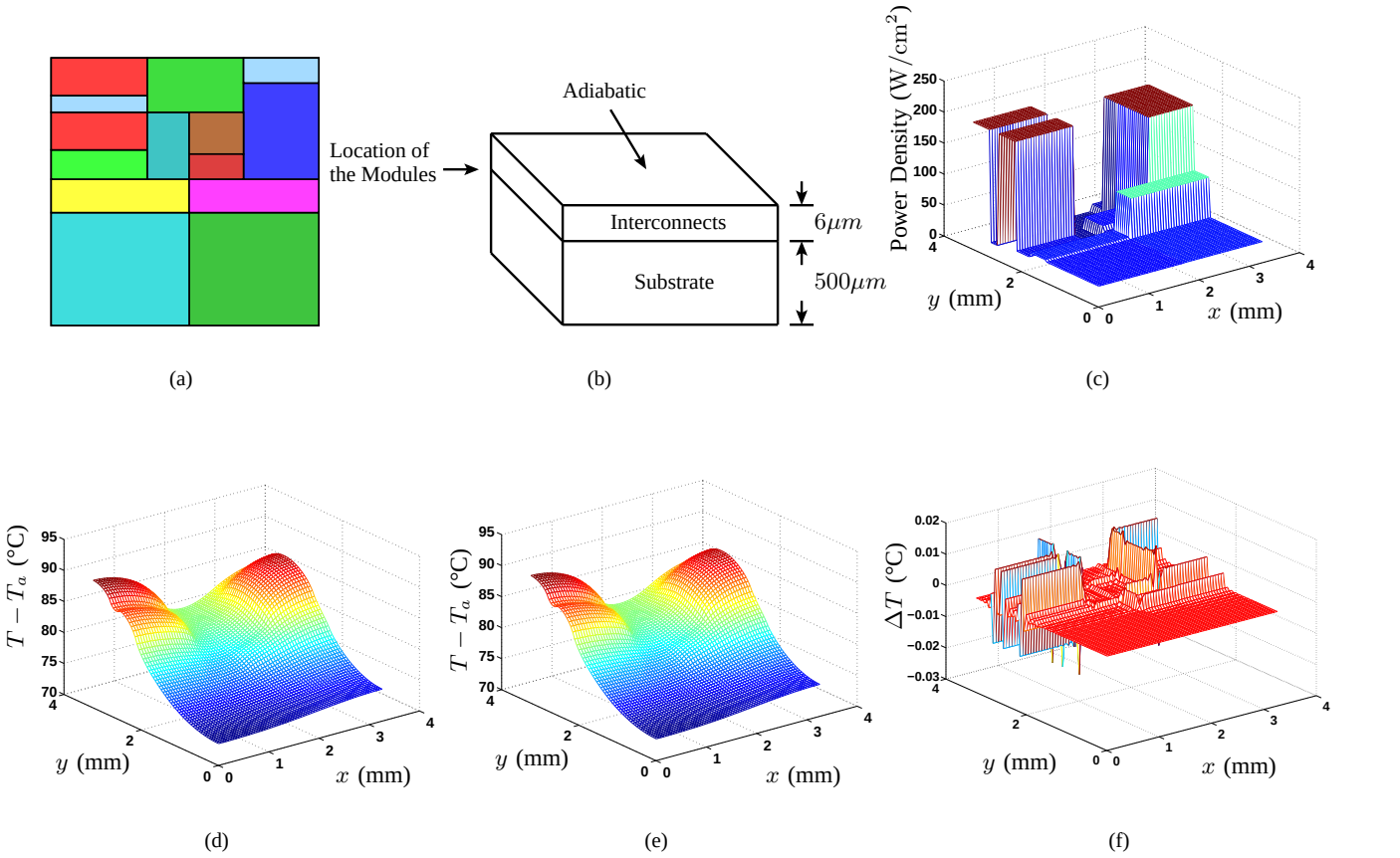
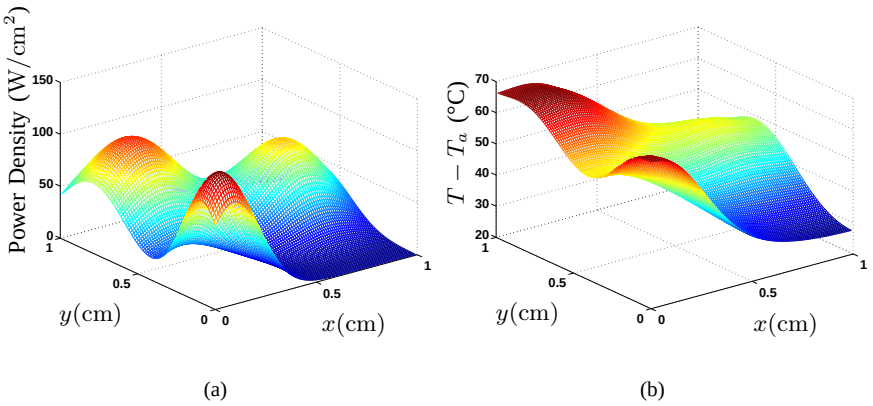


Fig. 10. Power and temperature distribution of a realistic chip (a) floorplan (b) schematic of the substrate and interconnect layers (c) power distribution (d) temperature distribution obtained using Algorithm I (e) temperature distribution obtained using Algorithm II (f) difference in the temperature distribution map obtained using the two algorithms.

interconnects is inserted between the insulating top surface and the substrate as shown in Fig 10(b). The added layer is assumed to have a thickness of 6μm and an effective thermal conductivity of 101W/(m·°C), which corresponds to a mixing of 25% copper, which has a thermal conductivity of 401W/(m·°C), and 75% oxide, which has a thermal conductivity of 1W/(m·°C). In real designs, the effective thermal conductivity of the interconnect layer can be estimated by taking the weighted average of the thermal conductivities of interconnect metal and oxide based on the designers' experiences on the interconnect densities of previous designs[2]. We further assume that the power is generated by the modules located at the interface between the interconnect layer and the substrate, and the temperature profile of this interface where the modules are located is calculated. Fig. 10(c) shows the power density distribution of the modules in $\mathrm{W/cm^2}$. We divided the module layer into 64×64 small square regions with equal size and computed the temperature maps using Algorithm I and II, which are shown in Fig. 10(d) and (e). Fig. 10(f) shows the difference between the temperature maps obtained using the two algorithms, and we can see that the results match each other very well. From the figures, we can also observe that the temperature maps are much smoother than the power density map. This can be explained by the relatively high thermal conductivity of the silicon substrate and the horizontal heat transfer [22]. For the CPU times required to obtain the temperature maps, Algorithm I uses 30*m*sec after the look-up table and vectors have been pre-calculated, while Algorithm II only uses 10*m*sec after the spectral responses of the linear system determined by the underlying Green function have been pre-calculated. Note that the runtime of Algorithm I is linear with respect to the number of heat sources and there are only 14 heat sources in the example shown here. For cell level full-chip simulations where the number of heat sources is significantly larger, the advantages of Algorithm II will become even more obvious. Therefore, we conclude that Algorithm II is more suitable for full-chip temperature profiling, where a large number of heat sources and field regions are involved.

To further demonstrate the efficiency of Algorithm II in full-chip thermal simulations, we tested a chip with dimensions of 1cm$\times1$cm$\times0.5$mm and has the same physical properties as the chips used in Section IV.A. There are 1024×1024 square grid cells of equal size located on the top surface of the chip and a 1024×1024 temperature distribution map of the cell layer is calculated. Fig. 11 shows the input power density map and

[2]For early stages of physical design where the detailed information about routing is usually unavailable, it is reasonable to use a uniform effective thermal conductivity to characterize the thermal property of each interconnect layer.



(a) (b)

Fig. 11. Cell level power density and temperature distribution of a 1cm×1cm chip (a) power density distribution (b) temperature distribution.

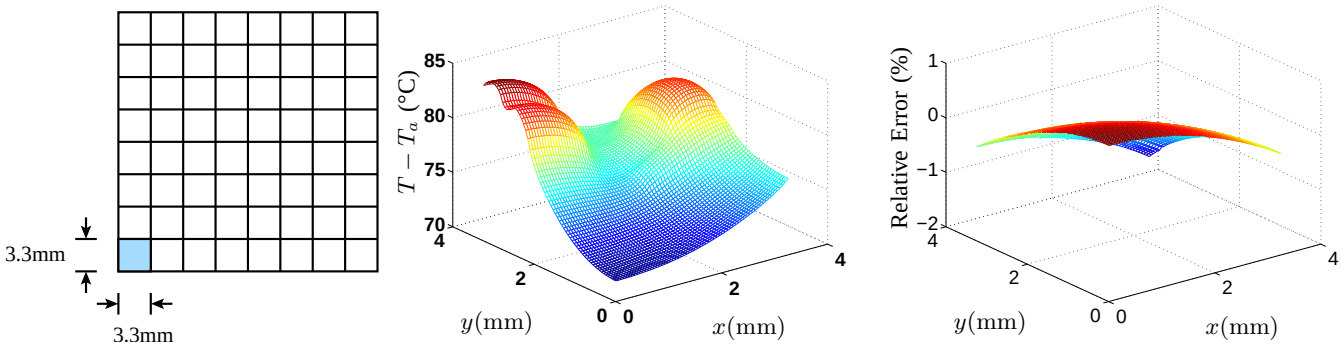
the resulting temperature map. The time it takes to obtain this temperature map containing 1.05M grid cells is only 3.7sec, excluding the time for the pre-calculations, while the runtime of Algorithm I becomes intractable.

C. Effectiveness of Algorithm III

Finally, we show an example of thermal simulation with local high accuracy requirement. We consider a chip that contains 8×8 coarse grid cells each of which has dimensions of 3.3mm×3.3mm, as shown in Fig. 12(a). The chip has the same material properties as the ones used in Section IV.A. We embed the layout and power density distribution shown in Fig. 10(a) and (c) in the coarse grid cell located at the lower left corner of the chip, which we denote by $CGC(0,0)$ in the following analysis, and the power density of each of the other 63 coarse grid cells is randomly generated between 0 and 100W/cm^2. Suppose that we want to obtain a 8×8 coarse temperature map over the 64 coarse grid cells and a 64×64 fine temperature map within $CGC(0,0)$. We compare two simulation schemes. In the first scheme, Algorithm II alone is used. In order to achieve the accuracy requirement of the fine temperature map within $CGC(0,0)$, we have to divide each of the 64 coarse grid cells into 64×64 fine cells, which results in a total of 512×512 fine cells over the entire chip. The time it takes to complete this simulation is 850*m*sec. In the second simulation scheme, we first obtain a 8×8 coarse temperature map from the 8×8 coarse power density map assuming that the power density within each coarse grid cell is uniform. The average temperature of $CGC(0,0)$ is found to be 79.4°C, while we know from the first simulation scheme that the actual temperature within $CGC(0,0)$ can vary from 71.2°C to 84.9°C. Next, we use a correction step as described in Section III.C to obtain the fine temperature map within $CGC(0,0)$. The overall runtime of the second simulation scheme for obtaining both the coarse and the fine temperature maps is only 70*m*sec, which is an order of magnitude faster than the direct application of Algorithm II with a fine grid over the entire chip. In Fig. 12(b) and (c), we show the fine temperature map within $CGC(0,0)$ achieved after the correction step and the relative error compared with the result obtained using the first simulation scheme. We can see that the maximum relative error is only about 1.3%. This demonstrates that using Algorithm III with a coarse grid and the correction scheme can indeed achieve a local accuracy comparable to that obtained by Algorithm II with a fine grid over the entire chip, while the overall runtime is significantly reduced.

V. Conclusions

In this paper, we presented three highly accurate thermal simulation algorithms based on the Green function method and analyzed in detail the relative advantages of each of the algorithms. Algorithm I combines the DCT and the table look-up technique to significantly reduce the time required for each evaluation of the Green function, and it is suitable for efficiently performing the localized analysis, where the effects of a few critical circuit blocks on the temperature distributions in a few field regions are sought. Algorithm II is based on the spectral domain analysis, and it takes advantage of the high efficiency of the FFT algorithm in transforming signals between the space and spectral domains. For full-chip thermal simulations, it can achieve an $O(\mathcal{N}_{gs} \times \log(\mathcal{N}_{gs})) + O(\mathcal{N}_{gf} \times \log(\mathcal{N}_{gf}))$ asymptotic time complexity as opposed to the $O(\mathcal{N}_{gs} \cdot \mathcal{N}_{gf})$ complexity of Algorithm I, where $\mathcal{N}_{gs}$ and $\mathcal{N}_{gf}$ are the total number of grid cells in the source and field planes, respectively. Algorithm III is a combination of both Algorithm I and Algorithm II, and it reflects the idea of the pre-corrected FFT. Its key application area is the full-chip thermal simulation with different accuracy requirements over the same chip, such as in the mixed signal design environments, where the analog blocks often have more stringent requirements on the accuracy of thermal simulations over the digital blocks. Experimental results show that all three algorithms can achieve around 1% errors compared with that of a commercial computational fluid dynamics software package for thermal analysis, while at the same time gaining orders of magnitude speedups over the classical Green function methods.



(a) (b) (c)

Fig. 12. Effectiveness of Algorithm III (a) location of the coarse grid cell $CGC(0,0)$ that has higher requirement on thermal simulation (b) temperature map within $CGC(0,0)$ calculated using Algorithm III (c) relative error of Algorithm III compared with Algorithm II applied with a fine grid over the entire chip.

REFERENCES

[1] D. Chen, E. Li, E. Rosenbaum, and S. M. Kang, "Interconnect Thermal Modeling for Accurate Simulation of Circuit Timing and Reliability," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 2, pp. 197-205, Feb. 2000.

[2] S. Rzepka, K. Banerjee, E. Meusel, and C. Hu, "Characterization of Self-heating in Advanced VLSI Interconnect Lines Based on Thermal Finite Element Simulation," *IEEE Transactions on Components, Packaging, and Manufacturing Technology, Part A*, vol. 21, no. 3, pp. 406-411, Sept. 1998.

[3] T. Y. Wang and C. P. Chen, "3-D Thermal-ADI: A Linear-Time Chip Level Transient Thermal Simulator," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 12, pp. 1434-1445, Dec. 2002.

[4] C. H. Tsai and S. M. Kang, "Cell-Level Placement for Improving Substrate Thermal Distribution," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 2, pp. 253-266, Feb. 2000.

[5] B. Goplen and S. S. Sapatnekar, "Efficient Thermal Placement of Standard Cells in 3D ICs Using a Force Directed Approach," *Digest of Technical Papers, 2003 IEEE/ACM International Conference on Computer-Aided Design*, pp. 86-89, Nov. 2003.

[6] P. Li, L. T. Pileggi, M. Asheghi, and R. Chandra, "Efficient Full-Chip Thermal Modeling and Analysis," *Digest of Technical Papers, 2004 IEEE/ACM International Conference of Computer-Aided Design*, pp. 319-326, Nov. 2004.

[7] A. Haji-Sheikh, "Peak Temperature in High-Power Chips," *IEEE Transactions on Electron Devices*, vol. 37, no. 4, pp. 902-907, Apr. 1990.

[8] Y. K. Cheng and S. M. Kang, "An Efficient Method for Hot-Spot Identification in ULSI Circuits," *Digest of Technical Papers, 1999 IEEE/ACM International Conference on Computer-Aided Design*, pp. 124-127, Nov. 1999.

[9] B. Wang and P. Mazumder, "Fast Thermal Analysis for VLSI Circuits via Semi-analytical Green's Function in Multi-layer Materials," *Proceedings of the 2004 IEEE International Symposium on Circuits and Systems*, pp. 409-412, May 2004.

[10] R. Gharpurey and R. G. Meyer, "Modeling and Analysis of Substrate Coupling in Integrated Circuits," *IEEE Journal of Solid-State Circuits*, vol. 31, no. 3, pp. 344-353, Mar. 1996.

[11] A. M. Niknejad, R. Gharpurey, and R. G. Meyer, "Numerically Stable Green Function for Modeling and Analysis of Substrate Coupling in Integrated Circuits," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 4, pp. 305-315, Apr. 1998.

[12] J. P. Costa, M. Chou, and L. M. Silveira, "Efficient Techniques for Accurate Modeling and Simulation of Substrate Coupling in Mixed-Signal IC's," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 18, no. 5, pp. 597-607, May 1999.

[13] J. R. Phillips and J. K. White, "A Precorrected-FFT Method for Electrostatic Analysis of Complicated 3-D Structures," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 16, no. 10, pp. 1059-1072, Oct. 1997.

[14] H. Hu, D. T. Blaauw, V. Zolotov, K. Gala, M. Zhao, R. Panda, and S. S. Sapatnekar, "Fast On-Chip Inductance Simulation using a Precorrected-FFT Method," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 22, no. 1, pp. 49-61, Jan. 2003.

[15] J. P. Costa, M. Chou, and L. M. Silveira, "Precorrected-DCT Techniques for Modeling and Simulation of Substrate Coupling in Mixed-Signal IC's," *Proceedings of the 1998 IEEE International Symposium on Circuits and Systems*, pp. 358-362, Jun. 1998.

[16] M. N. Ozisik, "Boundary Value Problems of Heat Conduction," Oxford University Press, Oxford, UK, 1968.

[17] A. G. Kokkas, "Thermal Analysis of Multi-Layer Structures," *IEEE Transactions on Electron Devices*, vol. 21, no. 11, pp. 674-681, Nov. 1974.

[18] Y. K. Cheng, P. Raha, C. C. Teng, E. Rosenbaum, and S. M. Kang, "ILLIADS-T: An Electrothermal Timing Simulator for Temperature-Sensitive Reliability Diagnosis of CMOS VLSI Chips," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 8, pp. 668-681, Aug. 1998.

[19] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, "Discrete-Time Signal Processing," Prentice Hall, Upper Saddle River, NJ, 1999.

[20] R. Gharpurey, "Modeling and Analysis of Substrate Coupling in Integrated Circuits," Ph. D. Thesis, UC Berkeley, Berkeley, CA, 1995.

[21] W. Liao, L. He, and K. Lepak, "Temperature-Aware Performance and Power Modeling," Technical Report UCLA Engr. 04-250, UCLA, Los Angeles, CA, 2004.

[22] K. Skadron, M. R. Stan, W. Huang, and S. Velusamy, " Temperature-Aware Microarchitecture," *Proceedings of the 30th International Symposium on Computer Architecture*, pp. 2-13, Jun. 2003.