

Improved pre-characterization method for the random walk based capacitance extraction of multi-dielectric VLSI interconnects

Bolong Zhang, Wenjian Yu^{*,†} and Chao Zhang

Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China

ABSTRACT

Accurately extracting capacitances among the interconnects in modern multi-dielectric technology of integrated circuits is a challenging task. The floating random walk (FRW) algorithm is advantageous for capacitance extraction and has been extended by W. Yu *et al.* to accurately handle multi-dielectric structures with a pre-characterization approach. In this paper, we improve this pre-characterization approach by permitting the cubic transition domains with three or four-dielectric layers. The techniques of pre-characterizing and utilizing these multi-dielectric transition cubes are proposed. Experiments on the test cases under actual manufacture technologies show that the proposed method brings 13× speedup on average, with affordable memory overhead. The experiments also validate the accuracy of the proposed method and reveal the significant error caused by the dielectric homogenization approach in a commercial FRW solver. Finally, on a machine with 12-core CPU, the parallel FRW algorithm equipped with the proposed pre-characterization method demonstrates more than 10× speedup of parallelization. Copyright © 2015 John Wiley & Sons, Ltd.

Received 11 June 2014; Revised 24 October 2014; Accepted 22 November 2014

KEY WORDS: capacitance extraction; floating random walk algorithm; improved pre-characterization; multi-dielectric; parallel computing

1. INTRODUCTION

As the feature size of integrated circuit technology decreases and the number of transistors increases, interconnect capacitance has a growing impact on circuit performance. Therefore, effective algorithms to extract the capacitance of interconnect conductors are crucial in the design of high-performance integrated circuits. The conventional deterministic algorithms, such as the boundary element method (BEM) [1–3], are fast and accurate but not suitable for large-scale structures because of the large amount of computational time and bottleneck of memory usage.

The floating random walk (FRW) algorithm for capacitance extraction, presented as a 2D version, was proposed in 1992 [4]. Its basic idea is to convert the calculation of electric potential to the FRWs in dielectric space. In the FRW algorithm, each walk starts from a point on the Gaussian surface enclosing the master conductor and terminates on a conductor surface after some successive hops. For each hop of a walk, a conductor-free square (or cube in 3D problem) centered at current location is constructed, and a random point on the square's boundary is selected. The spatial transition of hop obeys a probability distribution (called surface Green's function) on the square boundary. In a general 3D problem, if the cubic transition domain encloses a homogenous dielectric, the surface Green's function for the hop can be calculated analytically [5].

The FRW algorithm does not rely on assembling any linear equation system and has a variety of computational advantages over the deterministic methods: lower memory usage, more scalability for

*Correspondence to: W. Yu, Department of Computer Science and Technology, Tsinghua University, Beijing 100084 China.

†E-mail: yu-wj@tsinghua.edu.cn

large structures, tunable accuracy, and better parallelism. The FRW algorithm for capacitance extraction has been developed and applied to the design and analysis of VLSI circuits [5, 6]. In 2005, Batterywala *et al.* proposed several techniques to reduce the variance of Monte Carlo (MC) procedure in FRW-based capacitance extraction [7]. The FRW algorithm is able to handle multi-dielectric structure by introduction of sphere transition domain [8]. For actual VLSI interconnect embedded in up to 10 layers of dielectrics, this strategy will largely sacrifice the efficiency because the walk stops frequently at the dielectric interface. Because the transition probability for a cubic transition domain with inhomogeneous dielectric cannot be derived analytically, an approach is to numerically characterize the surface Green's function for the cubic transition domain with two-dielectric layers [9]. With this approach, the surface Green's function and weight value for the two-dielectric cubic transition domains are calculated and tabulated offline, and then recalled during the random walks. This pre-characterization method has been adopted in an FRW-based capacitance extractor RWCap [9, 10]. However, this method is still less efficient for actual multi-dielectric process technologies of VLSI circuits. Because each FRW hop in [9] only crosses one dielectric interface at most, quite a lot of hops are required when handling the actual VLSI technology with over ten dielectric layers. The aim of this work is to improve the pre-characterization method in [9], so that for each FRW hop the cubic transition domain can include three or four-dielectric layers. This will remarkably increase the runtime efficiency without sacrificing the accuracy. The techniques of pre-characterizing and utilizing the multi-dielectric transition cubes under a limited memory budget are proposed. Numerical experiments show that the techniques bring large speedup with affordable memory overhead.

The structure of this paper is as follows. Some basic knowledge of the FRW algorithm and numerical method to compute multi-dielectric surface Green's function and weight value are introduced in Section 2. An improved pre-characterization method is presented in Section 3. Numerical results to validate the efficiency of the FRW algorithm with the improved pre-characterization method are shown in Section 4, along with the comparison with other counterparts. Finally, the conclusion is presented in Section 5.

2. BACKGROUND

2.1. The floating random walk algorithm

The basic formula of the FRW algorithm is

$$\phi(\mathbf{r}) = \oint_S P(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)}, \quad (1)$$

where $(\phi\mathbf{r})$ is the electric potential at point \mathbf{r} , S is a surface surrounding the point \mathbf{r} , and $P(\mathbf{r}, \mathbf{r}^{(1)})$ is called surface Green's function. For a fixed point \mathbf{r} , $P(\mathbf{r}, \mathbf{r}^{(1)})$ can be regarded as the probability density function for selecting a random point $\mathbf{r}^{(1)}$ on S . In this sense, $(\phi\mathbf{r})$ can be estimated by the mean value of $\phi(\mathbf{r}^{(1)})$, if sufficiently large number of random samples are evaluated. If $\phi(\mathbf{r}^{(1)})$ is unknown, we apply (1) recursively to obtain the following formula

$$\phi(\mathbf{r}) = \oint_{S^{(1)}} P^{(1)}(\mathbf{r}, \mathbf{r}^{(1)}) \oint_{S^{(2)}} P^{(2)}(\mathbf{r}^{(1)}, \mathbf{r}^{(2)}) \dots \oint_{S^{(k+1)}} P^{(k+1)}(\mathbf{r}^{(k)}, \mathbf{r}^{(k+1)}) \phi(\mathbf{r}^{(k+1)}) d\mathbf{r}^{(k+1)} \dots d\mathbf{r}^{(2)} d\mathbf{r}^{(1)}, \quad (2)$$

where $S^{(i)} (i=1, 2, \dots, k+1)$ is the surface of the i -th cubic transition domain centered at $\mathbf{r}^{(i-1)}$. $P^{(i)}(\mathbf{r}^{(i-1)}, \mathbf{r}^{(i)})$ ($i=1, 2, \dots, k+1$) are the surface Green's functions relating the potentials at $\mathbf{r}^{(i-1)}$ to $\mathbf{r}^{(i)}$. The recursive expansion of integral in (2) terminates when the potential at point $\mathbf{r}^{(k+1)}$ is known. This successive spatial sampling corresponds to a random walk with several hops.

To extract capacitances among conductors, we can calculate it through the charge of conductors. With Gaussian theorem, the charge Q_i can be computed by the formula

$$Q_i = \oint_{G_i} F(\mathbf{r})g \oint_{S^{(1)}} \omega(\mathbf{r}, \mathbf{r}^{(1)}) P^{(1)}(\mathbf{r}, \mathbf{r}^{(1)}) \phi(\mathbf{r}^{(1)}) d\mathbf{r}^{(1)} d\mathbf{r}, \quad (3)$$

where G_i is Gaussian surface constructed to enclose the conductor i , and the weight value can be calculated by

$$\omega(\mathbf{r}, \mathbf{r}^{(1)}) = \frac{\nabla_r P^{(1)}(\mathbf{r}, \mathbf{r}^{(1)}) \cdot \hat{\mathbf{n}}(\mathbf{r})}{gP^{(1)}(\mathbf{r}, \mathbf{r}^{(1)})}, \quad (4)$$

where ∇_r is the gradient operator with respect to \mathbf{r} . $\hat{\mathbf{n}}(\mathbf{r})$ is the normal direction of G_i at \mathbf{r} . $F(\mathbf{r})$ is the dielectric permittivity at point \mathbf{r} . With MC method, Q_i can be estimated by the mean value of $\omega(\mathbf{r}, \mathbf{r}^{(1)})$ by sampling on Gaussian surface G_i with probability density function $F(\mathbf{r})g$. If $\phi(\mathbf{r}^{(1)})$ is unknown, we substitute (1) and (2) into (3) until $\phi(\mathbf{r}^{(k+1)})$ is known (i.e., $\mathbf{r}^{(k+1)}$ touches one of conductors). This may include several hops in a random walk. The walk starts from G_i and terminates when the hop touch conductor. It is revealed that the statistical mean of the weight values will approximate capacitance C_{ij} between conductor i and conductor j , when these corresponding walks terminate on conductor j .

For single-dielectric problem, the surface Green's function of cubic transition domain can be derived analytically, and pre-calculated and stored as the discrete probabilities for jumping to the cube surface [5]. For multi-dielectric problem, the pre-characterization method was proposed [9], where the surface Green's function and weight value for the cubic transition domain with two-dielectric layers are calculated numerically, and stored as Green's function tables (GFTs) and weight value tables (WVTs). Figure 1 shows an example of the FRW algorithm with the pre-characterization method in [9], where the cubic transition domain can cross one dielectric interface at most for each FRW hop.

Note that the total computing time of the FRW algorithm is roughly

$$T_{\text{total}} = N_{\text{walk}} \times N_{\text{hop}} \times T_{\text{hop}}, \quad (5)$$

where N_{walk} is the number of random walks, N_{hop} is the average number of hops in a walk, and T_{hop} is the average computing time for a hop.

2.2. Numerical technique to calculate multi-dielectric surface Green's function and weight value

The surface Green's function, which is the transition probability for selecting random point on the surface of the cubic transition domain, describes the relationship between the center point and boundary points of the cube, from the view point of potential. From formula (3), we can see that the weight value is the estimation of desirable capacitance, which is only related to sampling point on the first cube. With finite difference method (FDM), the surface Green's function and weight value for the cubic transition domain with two-dielectric layers can be calculated [9]. Actually, we can also numerically

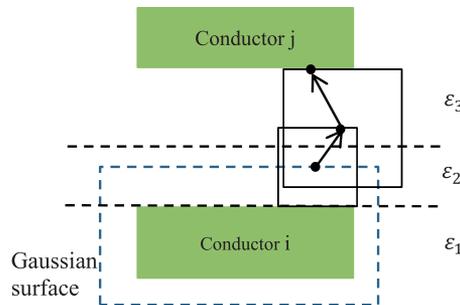


Figure 1. Example of the floating random walk algorithm with the pre-characterization method in [9].

compute surface Green's function and the weight value for the cubic transition domain with three or four-dielectric layers with FDM.

Suppose there is a cubic transition domain with four-dielectric layers, as shown in the Figure 2. In each homogeneous subdomain, the Laplace equation holds

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} = 0. \quad (6)$$

At the three-dielectric interfaces in Figure 2, there are boundary conditions, respectively,

$$\varepsilon_1 \frac{\partial \phi}{\partial z^-} = \varepsilon_2 \frac{\partial \phi}{\partial z^+}, \quad (7)$$

$$\varepsilon_2 \frac{\partial \phi}{\partial z^-} = \varepsilon_3 \frac{\partial \phi}{\partial z^+}, \quad (8)$$

$$\varepsilon_3 \frac{\partial \phi}{\partial z^-} = \varepsilon_4 \frac{\partial \phi}{\partial z^+}, \quad (9)$$

where $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ are the permittivities from low to high. Using FDM, the following matrix equation is derived from (6)–(9)

$$\begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} & \mathbf{E}_{13} \\ \mathbf{O} & \mathbf{I}_2 & \mathbf{O} \\ \mathbf{E}_{31} & \mathbf{O} & \mathbf{D}_{33} \end{bmatrix} \begin{bmatrix} \phi_I \\ \phi_B \\ \phi_F \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{f}_B \\ 0 \end{bmatrix}, \quad (10)$$

where ϕ_I, ϕ_B, ϕ_F are the potential unknown on inner grid points, surface panels, and dielectric interfaces, respectively. \mathbf{f}_B is the vector of the boundary potentials, which is as the Dirichlet boundary condition. Compared with two-dielectric situation, \mathbf{E}_{11} is changed at the locations where the corresponding panels touch new introduced interfaces, and the dimension of \mathbf{E}_{13} increases due to the new introduced interface elements in interfaces. After discretization, the cube surface is dissected into small panels. The transition probability we want is the relationship of the potentials at the center point and the surface panels. Then, we can compute surface Green's function by following formula

$$\mathbf{P}_k = -\left((\mathbf{E}_{11} - \mathbf{E}_{13} \mathbf{D}_{33}^{-1} \mathbf{E}_{31})^{-1} \mathbf{e}_k \right)^T \mathbf{E}_{12}, \quad (11)$$

where \mathbf{e}_k denotes the vector where the k -th element is 1 and otherwise are 0, and k is the index of the center point of cubic transition domain in the discrete inner grid points.

To calculate weight value, we will first drive following formula according to (4)

$$\omega(\mathbf{r}, \mathbf{r}^{(1)}) = -\frac{1}{g \cdot L} \cdot \frac{\frac{\partial P(\mathbf{r}, \mathbf{r}^{(1)})}{\partial x} n_x + \frac{\partial P(\mathbf{r}, \mathbf{r}^{(1)})}{\partial y} n_y + \frac{\partial P(\mathbf{r}, \mathbf{r}^{(1)})}{\partial z} n_z}{P(\mathbf{r}, \mathbf{r}^{(1)})}, \quad (12)$$

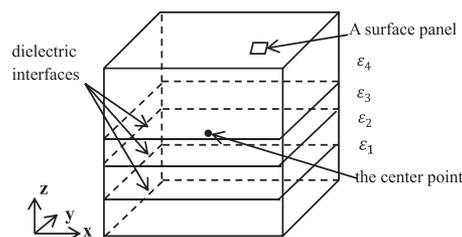


Figure 2. Transition cube with four-dielectric layers.

where L is the edge length of the first cubic transition domain, and we assume $\hat{n}(\mathbf{r}) = [n_x, n_y, n_z]^T$. Then we apply the finite difference formula for partial derivative in (12) to calculate weight value. More details can be found in [9].

3. AN IMPROVED PRE-CHARACTERIZATION AND EXTRACTION METHOD

In this section, we first introduce a dielectric homogenization approach used in Rapid3D [11]. Then, the method that improves the pre-characterization approach in [9] is proposed. After that, we describe the FRW algorithm with the proposed method.

3.1. The dielectric homogenization method

The basic idea of the dielectric homogenization method in [11] is as following. For a cubic transition domain with multiple dielectric layers (no matter how many dielectric layers it contains), it can be approximated by a cubic transition domain with four equal-thickness dielectric layers. An example is shown in Figure 3. The cubic transition domain with three-dielectric layers with thickness t_1, t_2, t_3 and permittivities $\epsilon_1, \epsilon_2, \epsilon_3$ from low to high is divided into four equal-thickness parts. The thickness of each part is $h = \frac{t_1+t_2+t_3}{4}$. Then, an equivalent permittivity of dielectric can be calculated for each part with a weighted average formula. Suppose ϵ_{part_i} , ($i = 1, 2, 3, 4$) denote the equivalent permittivity for the i -th dielectric part. Taking the second part in Figure 3 as an example, we have

$$\epsilon_{\text{part}_2} = \frac{t'_1 \epsilon_1 + t'_2 \epsilon_2}{h}, \tag{13}$$

where $t'_1 = t_1 - h$, $t'_2 = 2h - t_1$ are the thickness of different dielectrics in the second part. Therefore, the original cubic transition domain is approximated by the cubic transition domain with four equal-thickness dielectrics, whose permittivities are ϵ_{part_i} , ($i = 1, 2, 3, 4$). If with a pre-characterization approach we can pre-calculate the GFTs and WVTs for all configurations of the cubic transition domain with four-equal-thickness layers, they can be used to replace actual transition domains and release the constraint of FRW algorithm for handling multi-dielectric environment.

To enumerate all dielectric configurations for the transition cube with four-equal-thickness dielectrics, we notice that the GFT or WVT is actually determined by the ratios of dielectric permittivities in the four layers. For example, suppose $\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4$ are the permittivities for the four-dielectric layers, and $\epsilon_{\text{max}} = \max\{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4\}$. Then, the GFT calculated with ϵ_i , ($i = 1, 2, 3, 4$) will be the same as that for the transition cube whose four-dielectric permittivities are $\epsilon'_i = \frac{\epsilon_i}{\epsilon_{\text{max}}}$, ($i = 1, 2, 3, 4$). Based on this observation, we can only consider the dielectric permittivities valued in interval $(0, 1]$ in the pre-characterization procedure. And, they are assigned sample values, for example all integer multiples of a step s , to cover the interval $(0, 1]$. For each sample configuration, the GFT and WVT of the transition cube with four-equal-thickness dielectrics are generated.

With the dielectric homogenization method, during the FRW procedure the maximum cubic transition domain touching conductor can be used for each hop. Suppose a cubic transition domain is that

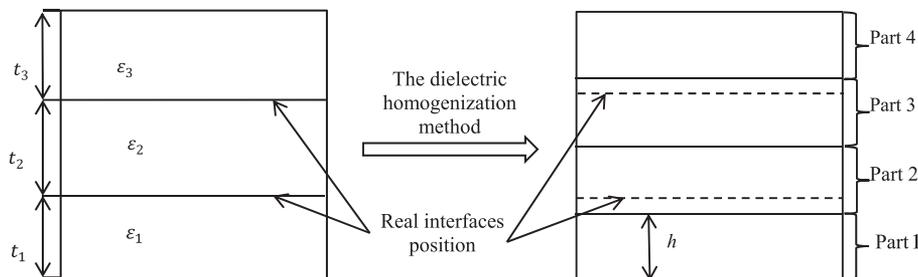


Figure 3. An example of the dielectric homogenization method. The cubic transition domain (cross-section view) is first divided into four equal-thickness parts, and then each part is assigned an equivalent permittivity, such as (13).

shown in Figure 3. First, the weighted average permittivities $\varepsilon_{\text{part}_1}, \varepsilon_{\text{part}_2}, \varepsilon_{\text{part}_3}, \varepsilon_{\text{part}_4}$ for the corresponding four-equal-thickness dielectrics are calculated. Then, their permittivity ratios are calculated and matched to a four-equal-thickness-dielectric cube, which has been pre-characterized. The permittivity ratio in the matched transition cube is calculated with

$$r_i = \left\lfloor \frac{\varepsilon_{\text{part}_i}}{\varepsilon_{\text{max}} \times s} + 0.5 \right\rfloor \times s, (i = 1, 2, 3, 4), \quad (14)$$

where $\varepsilon_{\text{max}} = \max\{\varepsilon_{\text{part}_i}\}, (i = 1, 2, 3, 4)$, and s is the step value for sweeping permittivity value in the aforementioned pre-characterization procedure. Finally, with the GFT and WVT of the matched transition cube the FRW hop can be executed.

We take the example in Figure 3 to explain the dielectric homogenization and the corresponding matching of the transition cube. Suppose the permittivities in the actual transition cube are $\varepsilon_1=2, \varepsilon_2=1, \varepsilon_3=3$, and the thickness of each dielectric layers are $t_1 = \frac{11}{36}, t_2 = \frac{13}{36}, t_3 = \frac{12}{36}$ (i.e., the heights of two-dielectric interface are $\frac{11}{36}$ and $\frac{24}{36}$). Then, the cubic transition domain will be matched by the pre-characterized transition cube with permittivities 0.65, 0.4, 0.45, and 1.

Figure 4 shows the comparison between the GFT distributions of the original cubic transition domain and the approximate transition cube with four equal-thickness dielectrics. From the picture, we see that the GFT distribution for the approximate transition cube is much different from the original GFT. This reveals the dielectric homogenization method may induce some error in handling the multi-dielectric structures. Our experimental results in Section 4 validate this analysis. In the next subsection, we will propose an improved pre-characterization method based on the pre-characterization method in [9], which will show good accuracy while bringing large speedup.

It should be pointed out that, there are two tricks to reduce the total data size for storing the GFTs and WVTs for all possible multi-dielectric configurations. (a) For actual process technology of VLSI, the value of each permittivity ratio is often in the interval $[0.5, 1]$. This means we may need fewer sample points to attain certain accuracy resolution. (b) The symmetry can be exploited in this pre-characterization procedure. For example, the GFTs and WVTs calculated with dielectric permittivities $(1, a, b, c)$ and $(c, b, a, 1)$ share the same GFT/WVT by rotating the transition cube.

3.2. The proposed method

3.2.1. Pre-characterizing the transition cube with three-layer or four-layer dielectrics. The basic idea of the proposed method is to pre-characterize two-dielectric, three-dielectric, four-dielectric GFTs and WVTs, so that the cubic transition domains are permitted to contain more than two-dielectric layers. This can be accomplished by the numerical technique in Section 2.2. In the

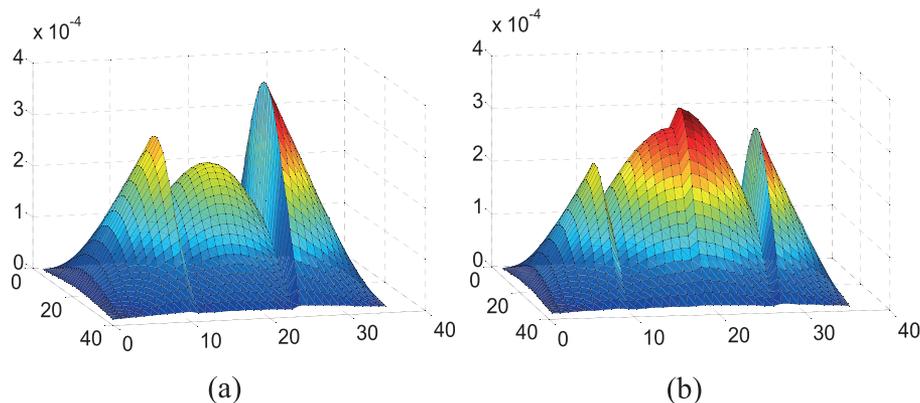


Figure 4. (a) Green's function table distribution of the unit transition cube with permittivities 2, 1, 3 from low to high and the height of each dielectric interface are $\frac{11}{36}, \frac{24}{36}$. (b) The Green's function table distribution.

pre-characterization, we consider a unit-size cubic transition domain, whose edge is divided into N segments.

In the pre-characterization of two-dielectric GFTs and WVTs [9], the dielectric interface is set at the heights of $\frac{i}{N}$, ($i = 1, 2, \dots, N - 1$) and $\frac{1}{2}$. And, the permittivities of the pair of dielectrics are set according to the given process technology. Then, for each configuration of interface height and dielectric permittivities FDM is employed to calculate the GFT and WVT. The same approach can be used to pre-characterize the three-dielectric or four-dielectric GFTs and WVTs. However, this will make the extra memory for the pre-characterized GFTs and WVTs exceeding 1 GB, even if only the three-dielectric transition cubes are considered [9].

For the sake of reducing the extra memory usage, we only allow the heights of dielectric interfaces in the three-dielectric or four-dielectric transition cubes to be $2 \times \frac{i}{N}$, ($i = 1, 2, \dots, \frac{N}{2} - 1$), instead of $\frac{i}{N}$, ($i = 1, 2, \dots, N - 1$). Note we suppose N is an even number. With this constraint there are only $\binom{N/2 - 1}{3}$ configurations for the dielectric interfaces in the pre-characterized four-dielectric transition cube. Because $4 \times 6N^2$ real numbers are needed to store the GFT and WVT for a dielectric configuration. If $N = 32$ and double-precision number is used, it needs 85 MB memory to store the GFTs and WVTs for the four-dielectric transition cubes with a given setting of dielectric permittivities. Under the same condition, the memory cost is 19.7 MB for the three-dielectric transition cubes. Compared with the strategy which allows the dielectric interface heights to be $\frac{i}{N}$, ($i = 1, 2, \dots, N - 1$), our approach reduces the memory cost by nearly 10 \times and 4.4 \times for pre-characterizing the four-dielectric and three-dielectric transition cubes, respectively.

A C++ program called TechGFT has been developed to pre-characterize the two-dielectric, three-dielectric, and four-dielectric transition cubes. The FDM techniques given in Section 2.2 and [9] are employed. Because the matrices involved in calculating the GFT and WVT, such as those in (11) are mostly sparse, UMFPACK [12] and CSparse [13] packages have been used to reduce the computational expense. Numerical experiments show that the TechGFT implemented in C++ is the same or more efficient than that implemented in Matlab [9].

3.2.2. Using the pre-characterized GFTs and WVTs during FRW procedure. The FRW algorithm using pre-characterized two-dielectric GFTs and WVTs has been presented in [9]. A shrinking operation for cubic transition domain may be performed, to adjust the size of the transition cube for a precise matching of the dielectric interface position with those in the pre-characterized cubes. This guarantees that the algorithm using two two-dielectric GFTs and WVTs induces no error. However, this shrinking operation does not work for the transition cubes with three or four-dielectric layers, because there are more than one dielectric interfaces, and it may be impossible to make all of them matching. In the succeeding text, we propose a shifting technique for utilizing the pre-characterized four-dielectric transition cubes during the FRW procedure. It makes the FRW hop across three-dielectric interfaces feasible, while inducing less error. Similar technique applies to the three-dielectric transition cube.

During the FRW procedure, we take the current position as the center and construct a transition cube including four-dielectric layers at most. Suppose there is such a four-dielectric transition cube whose size is r . $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$ are its dielectric permittivities from low to high. And, the heights of dielectric interfaces from low to high are h_1, h_2, h_3 . First, we correspond the transition cube to a unit-size transition cube. The heights of dielectric interfaces are normalized to $\tilde{h}_i = \frac{h_i}{r}$, ($i = 1, 2, 3$). Then, for matching the transition cube to the pre-characterized four-dielectric transition cubes, we shift each dielectric interface to the nearest dielectric interfaces candidates with heights $\frac{2}{N} \times i$, ($i = 0, 1, 2, \dots, \frac{N}{2} - 1$). This is accomplished by calculating:

$$t_i = \left\lfloor \frac{\tilde{h}_i \times N}{2} + 0.5 \right\rfloor, (i = 1, 2, 3). \quad (15)$$

Then, the i -th interface approaches to the t_i -th candidate value of dielectric interface height in the pre-characterized transition cubes. This means, the transition cube with dielectric interface heights

h_1, h_2, h_3 will be approximated by the pre-characterized four-dielectric transition cube with dielectric interface heights $2 \times \frac{t_i}{N}$, ($i = 1, 2, 3$). However, sometimes we may get $t_i = 0$ or $N/2$ (N is an even). And, $t_i = t_j$ may also hold for $i \neq j$. These situations all make the matched transition cube including fewer than four dielectrics. To avoid this error, minor adjustments of the dielectric interface positions are required. We propose some shifting operations to produce the valid matching to the four-dielectric transition cube. They are presented as Algorithm 1 in details.

Algorithm 1: The shifting operations for four-dielectric transition cube matching

- 1: Using 15 to get the number of dielectric interface position t_i , ($i = 1, 2, 3$)
- 2: **If** $t_i = 0$ or $N/2$, or $t_i = t_j$ ($i \neq j$) ($i, j = 1, 2, 3$)
- 3: $t_i = \min\{\max\{t_i, 1\}, \frac{N}{2} - 4 + i\}$, ($i = 1, 2, 3$) // adjust the top and bottom of the dielectric interface if $t_i = 0$, or $N/2$
- 4: **If** $t_1 = t_2 = t_3$ // three-dielectric interfaces coincide
- 5: $t_1 = t_1 - 1$; $t_3 = t_3 + 1$;
- 6: **Else if** $t_1 = t_2$ // two-dielectric interfaces coincide
- 7: **if** $t_1 + t_2 < t_3$
- 8: $t_2 = t_2 + 1$;
- 9: **else**
- 10: $t_1 = t_1 - 1$;
- 11: **end**
- 12: **Else if** $t_2 = t_3$ // two-dielectric interfaces coincide
- 13: **if** $t_2 + t_3 - t_1 < N/2$
- 14: $t_3 = t_3 + 1$;
- 15: **else**
- 16: $t_2 = t_2 - 1$;
- 17: **end**
- 18: **End**
- 19: **End**

The matching for three-dielectric transition cube can be similarly performed, but with less complexity than Algorithm 1. This is because only two-dielectric interfaces exist in the three-dielectric transition cube. To illustrate the shifting operations for matching the transition domain, we take a three-dielectric transition domain in Figure 4 as an example. The original transition cube and the matched transition cube are shown in Figure 5. In this example, the second dielectric interface precisely matches to the interface position in pre-characterized transition cube, while the first dielectric interface needs shifting before matching. For the matched transition cube, we draw its GFT distribution in Figure 6. Comparing Figure 6 with the original GFT in Figure 4(a), we see that the matching of transition cube causes less error. It is much more accurate than the GFT distribution got from the dielectric

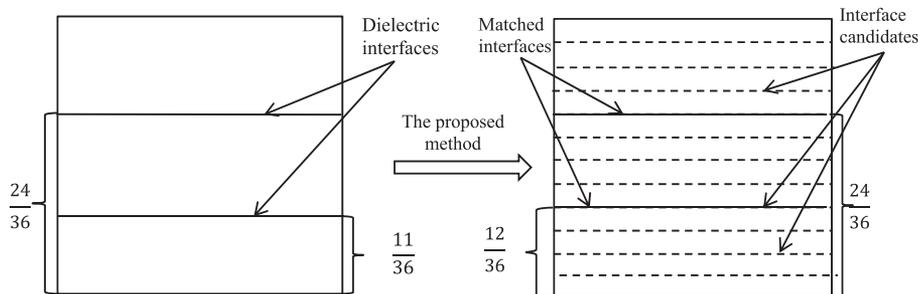


Figure 5. An example of three-dielectric transition cube in floating random walk procedure (same as that in Figure 4), and its matched peer in the pre-characterized transition cubes.

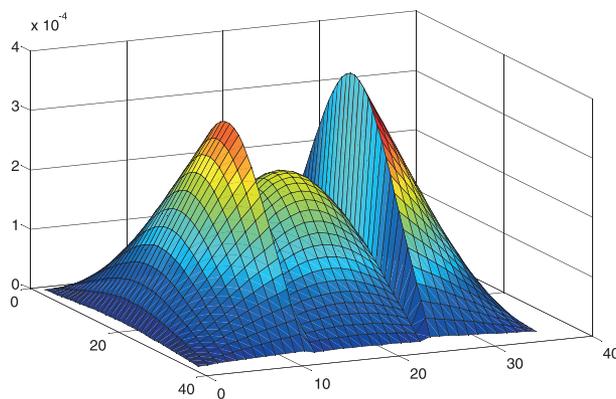


Figure 6. The Green's function table distribution of the matched transition cube with our approach, for the example in Figure 5.

homogenization method, that is, Figure 4(b). Therefore, the proposed pre-characterization approach and corresponding matching technique could preserve the accuracy of the FRW algorithm, and be advantageous over the dielectric homogenization method.

3.2.3. The FRW algorithm using three-dielectric and four-dielectric transition cubes. The FRW algorithm with two-dielectric pre-characterized GFTs and WVTs was introduced in [9], which limit cubic transition domain to contain two-dielectric layers at most. With the proposed techniques, we can pre-characterize three-dielectric and four-dielectric transition cubes and utilize them during the FRW procedure. Because the WVT is only useful for the first-hop transition domain, to reduce the memory cost of the pre-characterized GFTs and WVTs, we prohibit the first-hop transition domain including four-dielectric layers. Therefore, only GFTs are generated for the four-dielectric transition cubes. And, except the first hop, we still allow the transition cube to include four-dielectric layers. This makes a good tradeoff between the memory cost and the runtime efficiency. The improved FRW algorithm for handling multi-dielectric structures is described as Algorithm 2.

Algorithm 2: The improved FRW algorithm for handling multi-dielectric structures

- 1: For all dielectrics configurations occurring in the test structure, load two-, three-, and four-dielectric GFTs, and two- and three-dielectric WVTs;
- 2: Construct the Gaussian surface enclosing the master conductor j
- 3: $C_{ji} = 0, \forall i; n_k = 0, \forall k;$
- 4: **Repeat**
- 5: Pick a point $r^{(0)}$ on Gaussian surface, and then generate cubic transition domain T containing three-dielectrics layers at most; randomly pick a point $r^{(1)}$ on the T 's surface according to the pre-computed GFTs, and the calculate the weight value $\tilde{\omega}_k$ considering the position of $r^{(0)}$ and $r^{(1)}$;
- 6: **While** the current point is not on conductor surface **do**
- 7: Construct the largest conductor-free cubic domain containing at most four-dielectric layers;
- 8: Pick a point on the domain surface, according to the pre-characterized GFT of the cubic transition domain;
- 9: **End**
- 10: Register $\tilde{\omega}_k$ to strata k for C_{ji} ; //the current point is on conductor i
- 11: $n_k = n_k + 1;$
- 12: Calculate the standard error with the variance reduction technique in [9];
- 13: **Until** the stopping criterion is met
- 14: Calculate C_{ji} with the variance reduction technique in [9].

In steps 5 and 8, the shifting operations for matching the pre-characterized transition cubes should be performed. Then, with the GFT/WVT of the matched transition cube, the weight value and/or the transition probability can be used. Compared with the method in [9], the improved FRW algorithm would run faster with more memory cost. The numerical experiments in the next section will show the efficiency of the improved FRW algorithm for handling multi-dielectric structures.

4. NUMERICAL RESULTS

Two multi-dielectric interconnect structures are tested. The first one is a small case with three metal layers embedded in 12 planer dielectric layers. Figure 7 is the cross-section view of the process technology, where the relative permittivity of each dielectric layer is labeled. The case includes three parallel wires in the second metal layer M2. The width, height, and length of each wire are 70 nm, 140 nm, and 2000 nm, respectively. The second case is an actual design case called FreeCPU, which includes 37,062 conductor blocks in five metal layers embedded in 27 planer dielectric layers with 45 nm process technology. A portion of its layout is shown in Figure 8. Pre-characterized GFTs and WVTs have been generated by C++ program TechGFT, which takes 1.4s to calculate GFT formula (11) for one configuration and 3.8s to calculate the corresponding WVT. Total time used to produce GFTs, and WVTs for the process technology in Case 2 is about 3.87 h, which involves thousands of dielectric configurations. The FRW algorithm with the proposed method has been implemented in C++.

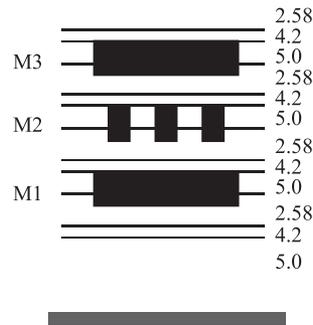


Figure 7. The cross section of process technology used in Case 1. The thickness of layers is 300, 40, 120, 100, 40, 120, 100, 40, 120, 100, 40, from bottom to top (in unit μm).

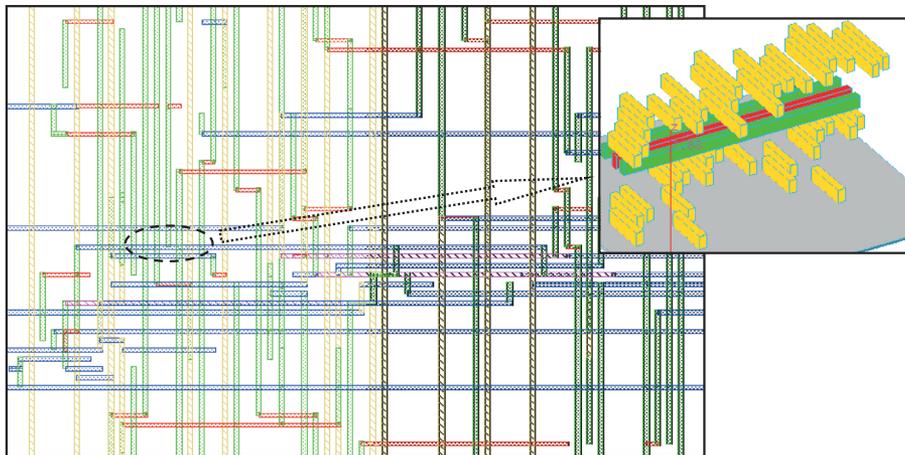


Figure 8. A small part of the 2D layout of Case 2 (called FreeCPU) and its zoom-in. The wires on different layers are distinguished by different color patterns.

Both TechGFT and the following experiments are carried out on a Linux server with Intel(R) Xeon (R) CPU E5-2630 of 2.30 GHz. The accuracy criterion of all FRW algorithms are set to 0.5% 1- σ error. Three versions of FRW algorithms are as follows.

RWCap:

The FRW algorithm of [9, 10], which utilizes the pre-characterized two-dielectric transition cubes.

RWCap(I):

The FRW algorithm with the proposed method (Section 3.2 and Algorithm 2).

RWCap(F):

The FRW algorithm using the pre-characterized cubes with four-equal-thickness dielectrics (Section 3.1).

4.1. Comparison with the original RWCap

Cases 1 and 2 have been tested by RWCap and RWCap(I). Results are obtained from the execution of serial computing. For Case 2, we take two conductors as the master conductor, respectively. Conductor 1 is in a middle metal layer, which is embedded in two planer dielectric layers. Conductor 2 is in the first metal layer, which is embedded in four-dielectric planer layers. Table I shows the computational results generated by RWCap and RWCap(I). From Table I, we see that compared with RWCap the memory usage of RWCap(I) increases for several times. It is 141 MB for Case 1 and 510 MB for Case 2. The increase of memory usage is due to the pre-characterized GFTs and WVTs for three-dielectric and four-dielectric transition cubes. Note that they are generated once for a process technology. While comparing the average number of hops per walk, RWCap(I) has remarkable advantage. This is the benefit of using four-dielectric transition cube, instead of using two-dielectric transition cube, during the random walk. In addition, for Case 2, if we take Conductor 2 as master conductor, the number of walks is reduced. This is because Conductor 2 is embedded in four-dielectric layers, such that the first transition cubes could be larger, crossing three-dielectric layers with the proposed method. From (12), we see the weight value ω is inversely proportional to the size of the first transition cube L . This larger first transition cube causes the reduction of variance of MC samples and therefore accelerates its convergence. In the experiments, we see that with the proposed method, the FRW algorithm is accelerated by 5.4 \times , 10 \times , and 37 \times , respectively, for the three tests.

For Case 2, we also randomly choose 100 nets as the master conductor and extract their capacitances. The runtime comparison is given in Table II. From Table II, we see the proposed method brings 13 \times speedup on average, for this 45 nm-technology design layout.

From Table I, we see the error of RWCap(I)'s result is -0.2% , -0.2% , and $+0.5\%$, respectively, if compared with RWCap's result. For further validation of the accuracy of RWCap(I), we run Case 1 with RWCap and RWCap(I) for 3000 times, respectively. Then, the distribution of extracted C_{self} is plotted in Figure 9(a-b). From the figure, we see that the plots both approximate the normal

Table I. Computational results of RWCap and RWcap(I) (Capacitance in unit of 10^{-16} F).

Case	RWCap					RWCap(I)							
	Hops	Walks	Mem.	C_{self}	Time	Hops	Walks	Mem.	C_{self}	Time	Sp.	Err(%)	
1	87.57	147 K	18	48.03	4.41	15.98	130 K	141	47.92	0.81	5.4	-0.2	
2	Cond. 1	152.9	4546 K	78	3.894	243.8	12.70	4476 K	510	3.886	24.7	9.9	-0.2
	Cond. 2	162.3	5438 K	78	1.88	329.9	13.50	1418 K	510	1.89	8.92	37	+0.5

*Hops means average number of hops per walk; Walks means the number of walks; C_{self} represents the self capacitance of master conductor; Mem. means memory usage in the program (in unit of MB). Time represents the computational time of the program (in unit of second).

Table II. Computational times of extracting 100 nets by RWCap and RWCap(I).

Case 2	Time(s)		Speedup
	RWCap	RWCap(I)	
100 nets	41248	3161	13

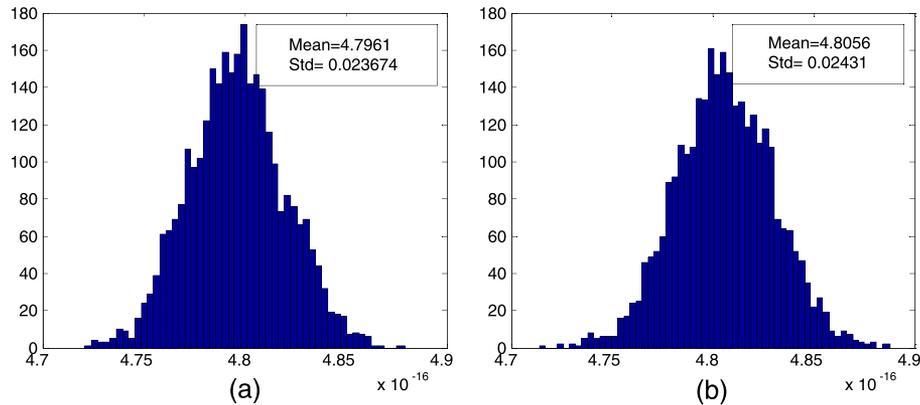


Figure 9. Distribution of C_{self} extracted with (a) RWCap and (b) RWCap(I).

distribution, with Std about 0.5% of the means value as prescribed. Comparing Figure 9(a-b), we find out that the mean value got by RWCap(I) is 0.2% larger than that by RWCap. This means that the proposed pre-characterization method induces negligible error.

4.2. Comparison with the dielectric homogenization method

For comparison, the results of capacitance extraction using RWCap(F) are also listed in Table III. From the table, we see that the memory usage of RWCap(F) is almost the same as RWCap(I). However, the error of self capacitances obtained from RWCap(F) are 2.1%, 1.9%, and 2.8%, respectively. It consistently overestimates the capacitance and has larger error than the proposed RWCap(I). To further

Table III. Computational results RWCap(F) (Capacitance in unit of 10^{-16} F).

Case	RWCap(F)						
	Hops	Walks	Mem.	C_{self}	Time	Sp.	Err(%)
1	8.08	105 K	511	49.17	0.50	8.8	+2.1
2	Cond. 1	6.28	960 K	511	3.967	4.74	+1.9
	Cond. 2	6.09	845 K	511	1.932	4.33	+2.8

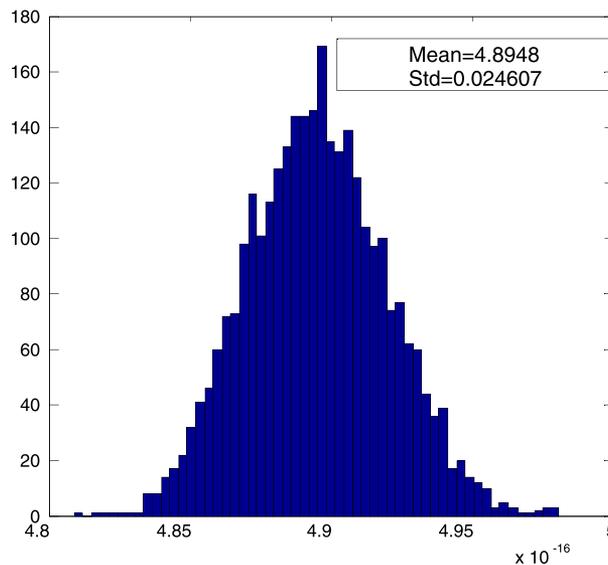


Figure 10. Distribution of C_{self} extracted with RWCap(F).

Table IV. Efficiency of the paralleled RWCap(I).

Case 2	# Thread	RWCap(I)	
		Time(s)	Speedup
100 nets	1	3161	—
	12	294	10.8

examine the accuracy of RWCap(F), we run Case 1 for 3000 times and then plot the distribution of extracted C_{self} in Figure 10. Comparing Figures 9(a) and 10, we see that the mean value got by RWCap(F) is about 2% larger. This reveals a significant error caused by the homogenization approach used in RWCap(F).

On the other hand, our experiment also shows that the dielectric homogenization method brings larger speedup to the capacitance extraction of multi-dielectric structure. This is because it permits the maximum-size transition cube for making the FRW hop. However, this speedup is at the cost of sacrificing accuracy.

4.3. Validating the efficiency of parallel computing

Parallel computing has been implemented in RWCap(I). On the 12-core machine, experiments are carried out to validate the efficiency of parallel computing with RWCap(I). The runtime for 100 nets in Case 2 are listed in Table IV. With 12 computing threads we achieve 10.8× speedup of parallelization. This reflects the high efficiency of parallelization of the proposed FRW extraction algorithm.

In our previous work [9], we have compared the FRW algorithm with the BEM-based fast solvers, like FastCap [1]. The results validate the accuracy of RWCap, and show that even for small multi-dielectric structure like Case1 FastCap is about 7× slower. Considering that RWCap(I) is 5.7× faster than RWCap for Case1, its speedup ratio to FastCap can be several tens. A distinct advantage of the FRW-based solver over the traditional solvers based on FDM, FEM, or BEM is that its runtime is almost independent to the problem size [14, 15]. Therefore, for larger interconnect structures, the proposed RWCap(I) solver should be more runtime and memory efficient than the traditional solvers involving matrix inversion.

5. CONCLUSION

The FRW algorithm with the improved pre-characterization method has been introduced in this paper. Compared with the existing pre-characterization method [9], the proposed method allows the FRW hop across up to four-dielectric layers and therefore exhibits faster computational speed. A shifting technique is proposed in the improved FRW algorithm to match the actual transition cube with a pre-characterized one, which preserves the accuracy as well. With about 500 MB memory cost, the proposed method becomes more than 10× faster than the method in [9] for actual structures under nanometer process technologies. Our experiments also reveal the significant error brought by the dielectric homogenization method [11] and show the better accuracy of the proposed method.

At last, it should be pointed out that if the single-precision float number is used to store the GFTs and WVTs, the memory cost of the proposed method would be cut by half without loss of accuracy.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support from the National Natural Science Foundation of China under contract no. 61422402, the Beijing Natural Science Foundation under contract no. 4132047, and the Tsinghua University Initiative Scientific Research Program.

REFERENCES

1. Nabors K, White J. Multipole accelerated capacitance extraction algorithm for 3-D structures with multiple dielectrics. *IEEE Trans Circuits Syst I* 1992; **39**(11): 946–954.
2. Shi W, Liu J, Kakani N, Yu T. A fast hierarchical algorithm for 3-D capacitance extraction, in *Proc. DAC*, 1998; pp. 212–217.

3. Yu W, Wang X, Ye Z, Wang Z. Efficient extraction of frequency dependent substrate parasitics using direct boundary element method. *IEEE Trans Comput-Aided Des* 2008; **27**(8):1508–1513.
4. Le Coz Y, Iverson RB. A stochastic algorithm for high speed capacitance extraction in integrated circuits. *Solid State Electron* 1992; **35**(7):1005–1012.
5. Iverson RB, Le Coz Y. A floating random-walk algorithm for extracting electrical capacitance. *Math Comput Simul* 2001; **55**:59–66.
6. Kamon M, Iverson R. High-accuracy parasitic extraction. In *EDA for IC Implementation, Circuit Design, and Process Technology*, Lavagno L, Scheffer L, Martin G (eds.). CRC Press/Taylor and Francis: Boca Raton, FL, 2006.
7. Batterywala SH, Desai MP. Variance reduction in Monte Carlo capacitance extraction, in *Proc. 18th Int. Conf. VLSI Design*, 2005; pp. 85–90.
8. Hu G, Yu W, Zhuang H, Zeng S. Efficient floating random walk algorithm for interconnect capacitance extraction considering multiple dielectrics. in *Proc. IEEE Int. Conf. ASIC*, 2011; pp. 896–899.
9. Yu W, Zhuang H, Zhang C, Hu G, Liu Z. RWCcap: a floating random walk solver for 3-D capacitance extraction of VLSI interconnects. *IEEE Trans Comput-Aided Des* 2013; **32**(3):353–366.
10. <http://learn.tsinghua.edu.cn:8080/2003990088/rwcap.htm> [Accessed 1 May 2014].
11. Rollins G. Rapid3D 20× performance improvement, <https://www.synopsys.com/Community/UniversityProgram/CapsuleModule/Rapid3D%2020X%20Performance%20Improvement.pdf> [Accessed 1 May 2014], 2010.
12. Davis TA. UMFPACK user guide. <http://www.cise.ufl.edu/research/sparse/umfpack/UMFPACK/Doc/UserGuide.pdf> [Accessed 1 May 2014].
13. Davis TA. *Direct Methods for Sparse Linear Systems*. SIAM Press: Philadelphia, 2006.
14. Le Coz Y, Greub HJ, Iverson RB. Performance of random walk capacitance extractors for IC interconnects: a numerical study. *Solid-State Electron* 1998; **42**(4):581–588.
15. Zhang C, Yu W. Efficient space management techniques for large-scale interconnect capacitance extraction with floating random walks. *IEEE Trans Comput-Aided Des* 2013; **32**(10):1633–1637.