Fast Floating Random Walk Algorithm for Multi-Dielectric Capacitance Extraction with Numerical Characterization of Green's Functions

Hao Zhuang[1,2], Wenjian Yu[1*], Gang Hu[1], Zhi Liu[1], Zuochang Ye[3]
[1] Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[2]School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China
[3]Institute of Microelectronics, Tsinghua University, Beijing 100084, China
[*]Tel: +86-10-6277-3440, Fax: +86-10-6278-1489, E-mail: yu-wj@tsinghua.edu.cn

Abstract–The floating random walk (FRW) algorithm has several advantages for extracting 3D interconnect capacitance. However, for multi-layer dielectrics in VLSI technology, the efficiency of FRW algorithm would be degraded due to frequent stop of walks at dielectric interface and constraint of first-hop length especially in thin dielectrics. In this paper, we tackle these problems with the numerical characterization of Green's function for cross-interface transition probabilities and the corresponding weight value. We also present a space management technique with Octree data structure to reduce the time of each hop and parallelize the whole FRW by multi-threaded programming. Numerical results show large speedup brought by the proposed techniques for structures under the VLSI technology with thin dielectric layers.

I. Introduction

As the feature size of IC technology decreases and the number of transistors increases, interconnect capacitance has a growing impact on circuit performance. Therefore, effective algorithms to extract the capacitances of interconnect conductors are crucial in the design of high performance integrated circuits. Traditional deterministic algorithms, such as the boundary element method with fast multi-pole acceleration [1] or quasi-multiple medium acceleration [13, 14], are fast and accurate, but not scalable to large structure due to the large demand of computational time or the bottleneck of memory usage.

A 2-D floating random walk (FRW) algorithm, which is discretization-free, was proposed to extract the interconnect capacitance [2]. The FRW algorithm is based on the Monte Carlo method for integral calculation, and converts the procedure of capacitance extraction to the random walks in dielectric space. It does not rely on assembling any linear systems of equation, and has a variety of computational advantages. Compared with the deterministic methods [1], the advantages include lower memory usage, more scalability for large structures, tunable accuracy, and better parallelism. The FRW algorithm has evolved to the commercial capacitance solver (such as *QuickCap* of Magma Inc.) for the design and analysis of VLSI circuits [3, 4], and has also some recent advances on the variance reduction technique [7] or for variation-aware capacitance extraction [5]. However, there is little literature which reveals the algorithm details of the 3-D FRW for multi-dielectric capacitance extraction.

The FRW algorithm is able to handle the multi-dielectric structure by introduction of sphere transition domain. For actual VLSI interconnects embedded in five to ten layers of dielectrics, this strategy will largely sacrifice the efficiency because the walk stops frequently at the dielectric interface. Another approach is to numerically generate the Green's function for the transition domain across dielectric interface. This was first seen in [6] for a Dirichlet problem (not capacitance extraction), where a FRW algorithm was used to generate the Green's function for sphere transition domain. It should be pointed out that the FRW slowly converges for generating the Green's function [6], and the sphere transition domain does not suit to the Manhattan shape of VLSI interconnects. This similar idea of generating Green's function was employed by the FRW-based capacitance extractor CAPEM [8]. However, the technical details of CAPEM are not published yet.

In this paper, the techniques based on finite difference method (FDM) are proposed to characterize the multi-dielectric Green's function and weight value for cubic transition domain. The pre-characterization procedure is compared with CAPEM, and exhibits speedup to the latter. Utilizing the multi-dielectric Green's function and weight value, the FRW is accelerated by several tens of times, with a little of memory overhead. To improve the FRW for structure with many conductors, a space management technique is proposed, which largely reduces the time of enquiring nearest conductors for each hop. Finally, the FRW algorithm is parallelized. The numerical results are carried out for 45nm-technology multi-dielectric structures, and the results including comparison with FastCap [1] and CAPEM [8] validate the accuracy and efficiency of proposed techniques. And, the experiments on a multi-core machine show that the paralleled FRW achieves more than 90% efficiency of parallelization.

II. Background

In the FRW algorithm, the fundamental formula is:

$$\Phi(r) = \oint_S P(r, r^{(1)}) \Phi\left(r^{(1)}\right) dr^{(1)} \quad , \qquad (1)$$

where $\Phi(r)$ is the electric potential on point r, S is a closed surface surrounding r. $P(r, r^{(1)})$ is called the Green's function. If S is the surface of a homogeneous cube or sphere centered at r, $P(r, r^{(1)})$ only depends on the relative position of $r^{(1)}$ with respect to r, and can be regarded as the probability density function (PDF) for selecting a random point on S. In this sense, $\Phi(r)$ can be estimated by $\Phi\left(r^{(1)}\right)$, if sufficiently large number of random samples are evaluated. In [2, 3], the Green's function for the cubic surface is derived analytically. And, the corresponding discrete probabilities for small pieces of cube surface are pre-computed, which forms the basis for an efficient FRW algorithm.

For the multi-conductor system within a single dielectric, (1) can be converted to a nested integral formula:

The work is supported by NSFC under Grant 61076034. W. Yu is the corresponding author.

978-1-4673-0772-7/12/$31.00 ©2012 IEEE

$$\Phi(r) = \oint_{S^{(1)}} P^{(1)}\left(r, r^{(1)}\right) dr^{(1)} \oint_{S^{(2)}} P^{(2)}\left(r^{(1)}, r^{(2)}\right) dr^{(2)} \cdots \oint_{S^{(k+1)}} P^{(k+1)}\left(r^{(k)}, r^{(k+1)}\right) \Phi\left(r^{(k+1)}\right) dr^{(k+1)} \quad , \qquad (2)$$

where $S^{(i)}, (i = 1, \cdots, k+1)$ is the ith cubic surface with center at $r^{(i)}$. $P^{(i)}, (i = 1, \cdots, k+1)$, are the Green's functions relating the potentials at $r^{(i-1)}$ and $r^{(i)}$. Once the potentials (voltages) on conductors are known, according to (2), the potential at r can be calculated with the following floating random walk procedure. Firstly, a maximum homogeneous cube centered at r is constructed, and a point $r^{(1)}$ is randomly selected on the cube surface $S^{(1)}$ according to the discrete probabilities resulted from the $P^{(1)}$ function. If $\Phi\left(r^{(1)}\right)$ is known (e.g. $r^{(1)}$ is on conductor surface), we can get an estimation of $\Phi(r)$. Otherwise, another cube centered at $r^{(1)}$ is constructed similarly, and $r^{(2)}$ is then randomly picked on $S^{(2)}$. These steps are repeated until $r^{(k+1)}$ is on conductor surface or remote from conductors, such that $\Phi\left(r^{(k+1)}\right)$ is known and becomes an estimation of $\Phi(r)$. This procedure is called a walk, for which the major cost is the geometric operation, since the probabilities are calculated in advance. After performing many walks, the mean value of these estimations becomes a fairly accurate $\Phi(r)$.

For extracting capacitances, the relationship between conductor charge and conductor potential is needed. So, a Gaussian surface G_i is constructed to enclose conductor i, and according to the Gauss theorem,

$$q_i = \oint_{G_i} D(r) \cdot \hat{n}(r) dr = \oint_{G_i} F(r)(-\nabla\Phi(r)) \cdot \hat{n}(r) dr, \quad (3)$$

where q_i is the charge on conductor i, D is the electric displacement, and $F(r)$ is the dielectric permittivity at point r. Substituting (2) into (3), we get:

$$q_i = \oint_{G_i} F(r) g \oint_{S^{(1)}} \omega P^{(1)}\left(r, r^{(1)}\right) \Phi\left(r^{(1)}\right) dr^{(1)} dr \quad , \quad (4)$$

where

$$\omega = -\nabla_r P^{(1)}\left(r, r^{(1)}\right) \cdot \hat{n}(r) / g P^{(1)}\left(r, r^{(1)}\right) \qquad (5)$$

is called the weight value, and constant g satisfies $\oint_{G_i} F(r) g dr = 1$. Now, the second integral in (4) can be calculated with the above FRW algorithm for potential, except for the extra calculation of ω. Similarly, the first integral in (4) also suggests a procedure of selecting points on G_i randomly. This results in a FRW algorithm for capacitance extraction [2, 3].

Though not explicitly presented in literature, the analytical Green's function for a sphere domain [9] can help the FRW to handle the situation with multiple dielectrics. Since the single-dielectric FRW relies on that the cube for point transition is in a single dielectric, a sphere transition domain can be used to continue the walk stopping at dielectric interface. This strategy is illustrated in Fig. 1(a), and described by the following Algorithm 1.

Algorithm 1 FRW for multi-dielectric problem

1: Load the pre-computed probabilities and weight values for single-dielectric cubic transition domain;
2: Construct the Gaussian surface enclosing master conductor *j*;
3: $C_{ji} \coloneqq 0, \forall i;\ npath \coloneqq 0;$
4: **Repeat**
5: $npath \coloneqq npath + 1;$
6: Pick a point $r^{(0)}$ on Gaussian surface, and then generate a single-dielectric cubic transition domain *T* centered at it; pick a point $r^{(1)}$ on the surface of *T*, and then calculate the weight value ω;
7: **Repeat**
8: **If** the current point is on dielectric interface, construct a sphere transition domain, **otherwise** construct a single-dielectric cubic domain;
9: Pick a point on the domain surface, according to the probabilities of cubic domain or sphere domain;
10: **Until** the current point touches a conductor *i*
11: $C_{ji} \coloneqq (C_{ji} * (npath - 1) + \omega)/npath$;
12: **Until** the stopping criterion is met

However, while extracting the actual VLSI interconnect structure with multiple dielectrics, Algorithm 1 would be very time consuming. If the Green's function, i.e. the probabilities, for a cube domain involving multiple dielectrics is available, the walk will cross dielectric interface. This can reduce the number of hops in a walk (e.g. see Fig. 1(b)), thus reduce the computing time of FRW.

The total computing time of FRW algorithm is roughly: $T_{total} = N_{walk} \times N_{hop} \times T_{hop}$. N_{walk} is the number of random walks/paths, N_{hop} is the number of hops in a walk, and T_{hop} is the time required for a hop. The techniques presented in the following two sections will reduce N_{hop}, N_{walk} and T_{hop} in turn.

III. Numerical Characterization of the Multi-Dielectric Green's Function and Weight Value

The Green's function describes the relationship between the center point and boundary points of a transition domain, from the view point of potential. This guides the selection of random point on boundary of transition domain. The weight is the estimation of desirable capacitance, and is only related with the transition in the first cube. In this section, the numerical techniques for calculating the Green's function and weight value for multi-dielectric cube are introduced.

A. Problem Formulation

In this work, we consider the unit-size cube domain including two dielectric layers (see Fig. 2 for cross-section view), and derive the Green's function and weight value. In

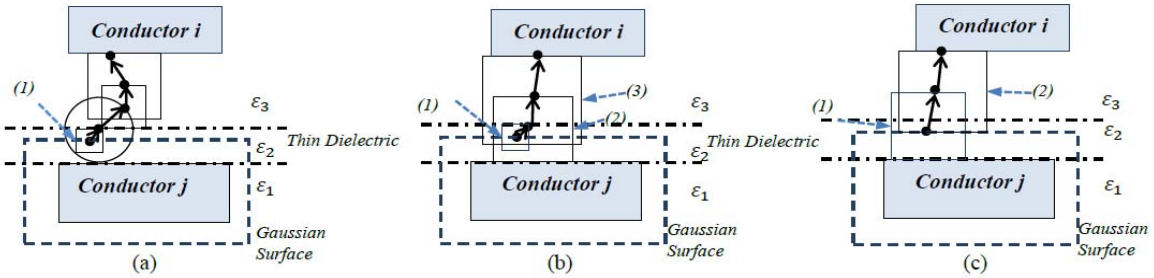

Fig 1. The random walk from conductor *j* to conductor *i* in multi-dielectric layers: (a) Algorithm 1, (b) using GFT, (c) Algorithm 2.

each homogeneous sub-domain, the Laplace equation holds:

$$\nabla^2\Phi = \frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2} + \frac{\partial^2\Phi}{\partial z^2} = 0 \quad . \tag{6}$$

At the dielectric interface, there is the continuous condition:

$$\varepsilon^+ \frac{\partial\Phi}{\partial z^+}(r) = \varepsilon^- \frac{\partial\Phi}{\partial z^-}\ (r) \quad , \tag{7}$$

where ε^+ and ε^- are the permittivities of up and down dielectric, respectively. The problem is to solve equations (6), (7), and obtain the probabilities and weight values.

B. Generating the Green's Function

The FDM can be used to solve the Laplace equation with multi-dielectric region. After discretization (as shown in Fig. 2), the following matrix equation is got:

$$\begin{bmatrix} \boldsymbol{E}_{11} & \boldsymbol{E}_{12} & \boldsymbol{E}_{13} \\ \boldsymbol{O} & \boldsymbol{I}_2 & \boldsymbol{O} \\ \boldsymbol{E}_{31} & \boldsymbol{O} & \boldsymbol{D}_{33} \end{bmatrix} \begin{bmatrix} \Phi_I \\ \Phi_B \\ \Phi_F \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{f}_B \\ \boldsymbol{0} \end{bmatrix} \tag{8}$$

where Φ_I, Φ_B and Φ_F are the potential unknowns on inner grid points, boundary, and interface respectively. $\boldsymbol{f}_B$ is the boundary potentials, which may be given as the Dirichlet boundary condition. The third block row of (8) is derived from (7), so that $\boldsymbol{D}_{33}$ is a diagonal matrix. Also note that $\boldsymbol{I}_2$ is an identity matrix. We then have:

$$\Phi_I = -(\boldsymbol{E}_{11} - \boldsymbol{E}_{13}\boldsymbol{D}_{33}^{-1}\boldsymbol{E}_{31})^{-1}\boldsymbol{E}_{12}\boldsymbol{f}_B \tag{9}$$

It expresses the relationship between the inner points and the boundary points. Suppose the center of the cube is the *k*th grid point of FDM. Then,

$$\Phi_k = \boldsymbol{e}_k^T\Phi_I = -((\boldsymbol{E}_{11} - \boldsymbol{E}_{13}\boldsymbol{D}_{33}^{-1}\boldsymbol{E}_{31})^{-T}\boldsymbol{e}_k)^T\boldsymbol{E}_{12}\boldsymbol{f}_B \quad . \tag{10}$$

The row vector

$$\boldsymbol{P}_k = -((\boldsymbol{E}_{11} - \boldsymbol{E}_{13}\boldsymbol{D}_{33}^{-1}\boldsymbol{E}_{31})^{-T}\boldsymbol{e}_k)^T\boldsymbol{E}_{12} \tag{11}$$

represents the discrete probabilities for transition from the center point to the boundary panels, and is what we want. It can be proved that $\|\boldsymbol{P}_k\|_\infty = 1$.

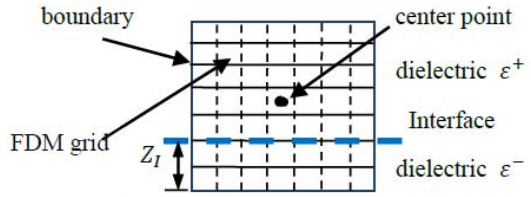

Fig 2. The cubic transition domain for calculating the Green's function (2D cross-section view) .

The above deduction assumes that both the center point and dielectric interface are adapted to the FDM grid. This requests that each edge of the domain is divided into *2N+1* segments, and one unknown is attached to each grid cell's center. They form the Φ_I for inner grid points. For each boundary panels, its center point is attached by an unknown, which forms Φ_B. Extra unknowns Φ_F are defined on the interface surface. Note that the height of interface Z_I has the value of $\frac{1}{2N+1}, \frac{2}{2N+1}, \cdots,$ or $\frac{2N}{2N+1}$ (see Fig. 2).

Different FD schemes are used to generate the coefficient matrix blocks in (8). For the inner grid point, the standard seven-point differential scheme is used (see Fig. 3(a)). For the grid point near boundary as shown in Fig. 3(b), its distance to boundary face is only h (we assume the inner grid size is $2h$). The Lagrange interpolation is used to derive the differential scheme. For example, we approximate the z-direction derivative for the point in Fig. 3(b) with:

$$\frac{\partial^2\Phi}{\partial z^2} \approx \frac{2u_0}{3h^2} - \frac{u_1}{h^2} + \frac{u_2}{3h^2} \tag{12}$$

Here u denotes the potential on grid point. For the point on

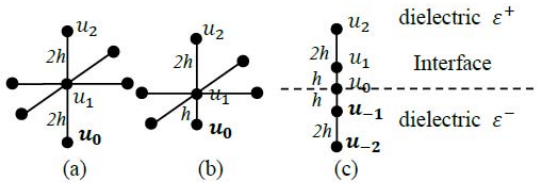

Fig. 3. Illustration of finite difference schemes for: (a) inner grid point, (b) grid point near boundary, (c) point on interface

interface, another FD equation is needed to approximate (7). In order to be consistent with the scheme for (6), the formula with second-order accuracy is required. As shown in Fig. 3(c), on each side of interface the Lagrange interpolation with three points is used to derive the approximate formula for $\frac{\partial\Phi}{\partial z^+}(r)$ or $\frac{\partial\Phi}{\partial z^-}(r)$. Finally, we get the following difference equation for the example in Fig. 3(c):

$$\varepsilon^+ \frac{-8u_0+9u_1-u_2}{6h} = \varepsilon^- \frac{8u_0-9u_{-1}+u_{-2}}{6h} \tag{13}$$

With this formula, the severe discontinuous problem for generated Green's function solutions can be avoided.

In the FDM solution, the height of interface Z_I cannot be 1/2, which omits an important situation like the cube (2) in Fig. 1(b). The special treatment should be taken. As shown in Fig. 4(a), which is a side view, the cube edge should be divided into even segments, so that there is no boundary panel across the interface. Therefore, the center point is not an inner grid point. As shown in Fig. 4(b), the potential of center point can be approximated by its eight neighbors. By modifying (11), an efficient approach is found to calculate the discrete probabilities for the transition cube whose dielectric interface is at height 1/2. Due to the limit of space, its detail is omitted here.

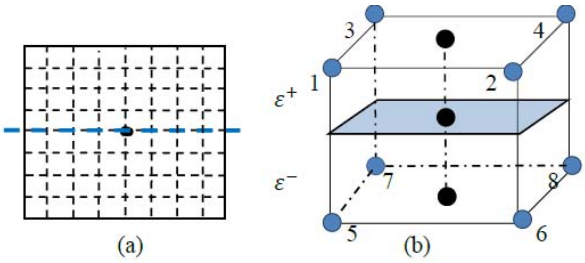

Fig. 4. The situation where the height of interface $\boldsymbol{Z_I}$=1/2.

The multi-layer Green's function can be pre-computed for all possible $(\varepsilon^+, \varepsilon^-)$ pairs. For each pair, a set of Green's function table (GFT) are generated for different interface positions. Fig. 5 shows the examples of GFTs for the situation: $\varepsilon^- = 2.6$, $\varepsilon^+ = 5$. Note that the picture in Fig. 5(a) is consistent with that of single dielectric Green's function in [3], and the plots in Fig. 5(b), (c) demonstrate that there is much larger probability to walk towards the dielectric with

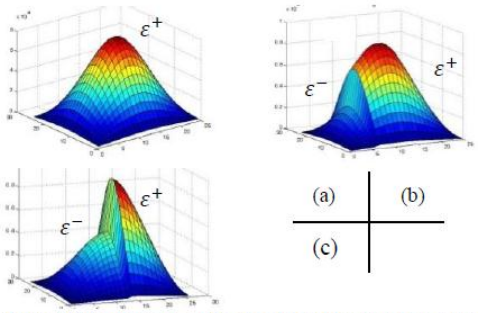

Fig. 5 The GFT plots on cube boundaries: (a) top face, (b) side walls with interface height of 1/5, and (c) 1/2. ($\varepsilon^- = 2.6$, $\varepsilon^+ = 5$)

higher permittivity. By setting $\varepsilon^- = \varepsilon^+$, we have used the single-dielectric Green's function to validate the accuracy of the GFTs obtained with the FDM.

C. Generating the Weight Values

To calculate the weight value in (5), we firstly derive:

$$\omega = -\frac{1}{gL}\left(\frac{\partial P/\partial x}{P}n_x + \frac{\partial P/\partial y}{P}n_y + \frac{\partial P/\partial z}{P}n_z\right), \quad (14)$$

where L is the size of the first cube, and P is the Green's function for the unit-size cube. With the technique in last subsection the values of P is available, and its parital derivatives can also be calculated with the FDM. For example, $\partial P/\partial x$ stands for the sensitivity of the probabilites with respect to the x-axis position of the center point of cube. Suppose n is the number of inner points per edge of cube. Since $\boldsymbol{P}_k$ in (11) is the GFT associated with the center point, we denote the GFTs associated with its six neighbor points to be $\boldsymbol{P}_{k+1}, \boldsymbol{P}_{k-1}, \boldsymbol{P}_{k+n}, \boldsymbol{P}_{k-n}, \boldsymbol{P}_{k+n^2}$ and $\boldsymbol{P}_{k-n^2}$. Then, with the centered difference formula we have:

$$\frac{\partial \boldsymbol{P}_k}{\partial x} \approx \frac{\boldsymbol{P}_{k+1}-\boldsymbol{P}_{k-1}}{2h} = -((\boldsymbol{E}_{11} - \boldsymbol{E}_{13}\boldsymbol{D}_{22}^{-1}\boldsymbol{E}_{31})^{-T}\tilde{\boldsymbol{e}}_1)^T\boldsymbol{E}_{12} \quad (15)$$

$$\frac{\partial \boldsymbol{P}_k}{\partial y} \approx \frac{\boldsymbol{P}_{k+n}-\boldsymbol{P}_{k-n}}{2h} = -((\boldsymbol{E}_{11} - \boldsymbol{E}_{13}\boldsymbol{D}_{22}^{-1}\boldsymbol{E}_{31})^{-T}\tilde{\boldsymbol{e}}_n)^T\boldsymbol{E}_{12} \quad (16)$$

$$\frac{\partial \boldsymbol{P}_k}{\partial z} \approx \frac{\boldsymbol{P}_{k+n^2}-\boldsymbol{P}_{k-n^2}}{2h} = -((\boldsymbol{E}_{11} - \boldsymbol{E}_{13}\boldsymbol{D}_{22}^{-1}\boldsymbol{E}_{31})^{-T}\tilde{\boldsymbol{e}}_{n^2})^T\boldsymbol{E}_{12} \quad (17)$$

Here vector $\tilde{\boldsymbol{e}}_i, (i = 1, n, n^2)$ is defined as:

$$\tilde{\boldsymbol{e}}_i(j) = \begin{cases} 1/2h & , j = k+i \\ -1/2h & , j = k-i \\ 0 & , otherwise \end{cases}. \quad (18)$$

Eq. (15)~(17) give the calucating formulas for the partial derivatives in (14). And for a small value of h, they would have sufficient accuracy. In practice, the values of $\frac{\partial P/\partial x}{P}$, $\frac{\partial P/\partial y}{P}$ and $\frac{\partial P/\partial z}{P}$ for each boundary panels (the position of $r^{(1)}$) are pre-computed and stored as the weight value table (WVT). As the same as the GFT, for each pair of dielectric permittivities a set of WVTs are needed for different interface positions.

With the WVTs, the first transition cube is not required to be within single dielectric (see Fig. 1(c) for an example). This enlarges the length of hop again with a little of memory overhead. Another noticeable advantage is that, by using WVT the MC procedure of FRW can converge much quickly. This is because ω is inversely proportional to the size of the first transition cube L, as shown in (14). Thus, increasing L largely reduces the probability of getting an unreasonably large MC sample value, especially for the situation there is a thin dielectric near the Gaussian surface (see Fig. 1). This will reduce the number of walks to attain a certain accuracy goal, therefore reduce the total computing time of FRW. The analysis is validated in Section V.

D. The FRW Algorithm with Multi-Dielectric GFT and WVT

The following algorithm describes the FRW algorithm utilizing the multi-dielectric GFT and WVT.

Algorithm2 FRW for multi-dielectric problem (with GFT and WVT)

1: (1) Load the pre-computed transition probabilities, weight values for single-dielectric cubic transition domain;
(2) Load the multi-dielectric GFTs and WVTs;
2: Construct the Gaussian surface enclosing master conductor j;
3: $C_{ji} \coloneqq 0, \forall i;\ npath \coloneqq 0$;
4: **Repeat**
5: $npath \coloneqq npath + 1$;
6: Pick a point $r^{(0)}$ on Gaussian surface, and then generate a cubic transition domain (may contain multiple dielectrics) T centered at it; pick a point $r^{(1)}$ on the surface of T, and then calculate the weight value ω according to the WVTs;
7: **Repeat**
8: Construct the largest cubic domain contains at most two dielectrics;
9: Shrink the domain a little to make the height of interface to match that in the GFTs, if needed;
10: Pick a point on the domain surface, according to the probabilities of cubic domain;
11: **Until** the current point touches a conductor i
12: $C_{ji} \coloneqq (C_{ji} * (npath - 1) + \omega)/npath$;
13: **Until** the stopping criterion is met

In the step 9 of Algorithm 2, a shrinking operation may be performed. Its aim is to adjust the size of transition cube to match its height of dielectric interface to that in the GFTs. This minimizes the numerical error induced by using GFTs.

The FDM-based method to generate the GFT and WVT is implemented in MATLAB, using the functions for sparse matrix. The proposed method is compared with CAPEM [8], which is a binary-coded program. For different segment numbers (2N+1) along one edge of cube, the computational time of both programs for each set of GFT and WVT are given in Fig. 6. From this figure, we can see that our MATLAB program is more efficient than CAPEM.

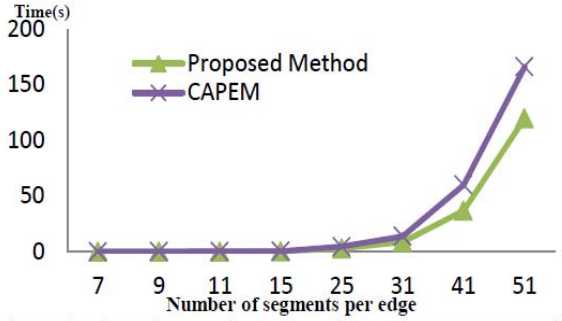

Fig 6. The time of generating GFT and WVT vs. segment number.

IV. Improving the Efficiency by Space Management and Parallel Computing

In this section, two techniques are presented to accelerate the FRW algorithm for capacitance extraction.

A. Space Management Technique

For each hop in the FRW algorithm, the distance from the current point to conductors should be measured for constructing the maximum transition cube. This is the procedure of finding the *nearest* conductor, whose time should be reduced as short as possible because millions of hops may be needed in the FRW algorithm. A space management technique is presented here to organize the conductors in the problem, such that the distance is calculated for only a few of conductors each time. Therefore, the computing time of each hop can be remarkably reduced, especially for the case involving many conductors. In this technique, the *Octree* spatial data structure [10] is used. Each *Octree* node represents a cubic spatial domain, which may be decomposed into eight subdomains and each one corresponds to a child node in the tree. For each node of *Octree*, we want to have a number possible nearest

conductors (candidate list) for its correspond domain. To build such an *Octree*, we firstly define the "*dominant*" relationship.

Definition 1: T is a cubic space, B_1 and B_2 are conductors. B_1 *dominate* B_2 regarding T, *iff*. $\forall P \in T$, $d(P, B_1) \le d(P, B_2)$, where $d(\ ,\ )$ is the function of the ∞-norm distance.

By inserting the conductors into a null *Octree*, the *Octree* can be constructed. Such an insertion procedure is described by Algorithm 3. A *threshold* is predefined as the maximum size of the candidate list.

Algorithm 3 Insert conductor B into node T

```
1:  If T is a leaf node then
2:    For each b in the candidate list of T do
3:      If b dominate B then return
4:      Elseif B dominate b then
5:        Remove b from T;
6:      Endif
7:    Endfor
8:    Add B to the candidate list of T;
9:    If the size of T's candidate list is larger than threshold then
10:     Divide T into 8 child nodes, and then redistribute its
        candidate list to them
11:   Endif
12: Else
13:   For each child c of T do
14:     Insert conductor B into node c
15:   Endfor
16: Endif
```

Once the *Octree* is established for the extracted structure, in each hop only the conductors in the candidate list of the leaf node that the current point belongs to are enquired to calculate distance. So, the time complexity of each hop is about $O(h+t)$, where h is the height of *Octree* and t is the threshold. We have tested two structures with 201 and 402 wires respectively. Numerical results show the running time of FRW algorithm is reduced by 72% or 88%, with negligible memory overhead.

B. The Parallel FRW Algorithm

The FRW algorithm is very suitable for parallelization, since the walks are independent to each other. Fig. 7 shows the flowchart of the parallel FRW algorithm on a multi-core/multi-CPU platform. Multiple threads are allocated to execute the random walk procedure (steps 6~11 in Algorithm 2; denoted by "FRW core" in Fig. 7). In order to reduce the expense of communication, a unique random number generator [11] is kept for each thread. The "lock" operation only happens when updating the value of capacitance and checking the program termination criteria with total number of walks or convergence test. This check is performed for every m walks (m=1000). The work of loading GFTs and WVTs is also parallelized without difficulty. Thus, only the works of parsing input, building the Octree and constructing the Gaussian surface are executed serially. The *pthread* APIs are used in our implementation.

V. Numerical Results

We have developed a C++ program RWCap with the proposed techniques. To evaluate the efficiency of different technique, three subversions of RWCap are defined:

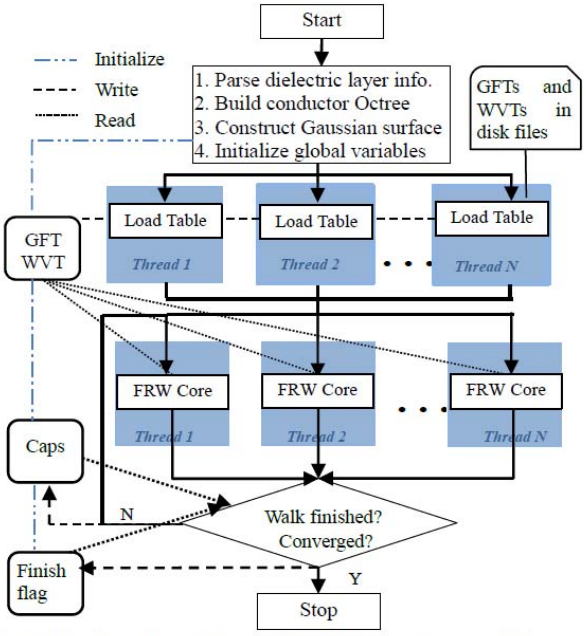

Fig. 7 The flowchart of the parallel FRW on multi-core platform.

- RWCap(O): The original multi-dielectric FRW (Algorithm 1).
- RWCap(G): The FRW using the multi-dielectric GFTs.
- RWCap(GW): The FRW using the multi-dielectric GFTs and WVTs (Algorithm 2).

The typical 45nm technology described in [12] is considered, and structures with three metal layers are tested. The cross section of the structures is shown in Fig. 8, where the relative permittivity of each dielectric layer is labeled. The stop criteria of RWCap is that the standard deviation of total capacitance estimations becomes smaller than 3% of the mean value. All experiments are carried out on a Linux server with Xeon E5620 8-core CPU of 2.40GHz.

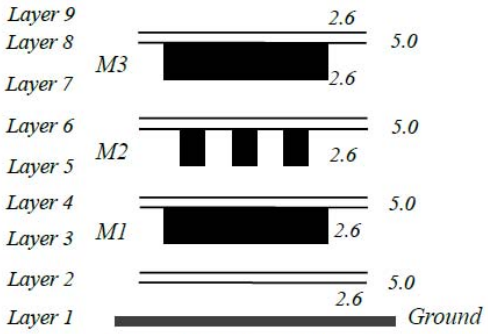

Fig. 8 The cross section of test structures under the 45nm technology. Layers 2, 4, 6, 8 are thin dielectrics with relative permittivity of 5.0 and thickness of 40nm. Other layers have permittivity of 2.6. The thickness of layers 3, 5 and 7 is 220nm.

A. Experiments of Serial Computing

The first case includes 3 parallel wires in M2 layer. The width, height and length of each wire are 70nm, 140nm and 2000nm, respectively. The wire spacing is 70nm. For the second case, two sets of 19 wires are added to M1 and M3 layers respectively, with random width and spacing. The results of RWCap for the two cases are listed in Table I.

TABLE I
The Computational Results of RWCap (O), (G) and (GW) for Case 1 and 2

RWCap	C_{11} (10^{-16} F)	# Walks	Memory (KB)	Time(s)	Speedup
Case 1 with 3 wires					
O	3.59	891891	1348	257.28	--
G	3.62	908908	5628	90.51	2.8
GW	3.61	22022	13808	2.03	127
Case 2 with 41 wires					
O	4.03	674674	1452	68.93	--
G	3.84	718718	5728	29.94	2.3
GW	3.86	19019	13892	0.78	88

From this table we can see that the technique of utilizing the multi-dielectric GFT speeds up the original FRW algorithm by about 2.5X. While using both GFT and WVT, the speedup increases to 127 and 88 for the two cases. And the total memory is only 13 MB, which is negligible. The larger acceleration brought by WVT is because it avoids the first transition domain with very small size, which makes the samples of MC procedure become fairly "uniform". This greatly reduces the number of walks for convergence. Case 1 is also solved with FastCap [1] and CAPEM [8]. Fastcap's result is 3.55×10^{-16} F, with time of 34.7 seconds and 1.8GB memory. It validates the accuracy of RWCap, and shows that FastCap is inferior even for the smaller case due to a lot of panels caused by discretization of dielectric interfaces. With 100K walks, CAPEM's result is 3.62±0.18 (5%), while consuming 773 seconds. To compare fairly, we also run RWCap(GW) for 100K walks, which produces C_{11} of 3.49± 0.045 (1.3%) after 14 seconds. This means RWCap(GW) is 55X faster than CAPEM and has better convergence rate.

Two more cases are then tested, and the results are given in Table II. Case 3 and 4 have the same structure of wires as Case 1 and 2, respectively, but they include 5 dielectric layers instead of 9 layers. This reduction is accomplished by merging the thin dielectric with one of its neighbors. With some empirical formula of equivalent permittivity, it may not degrade the accuracy a lot. For this experiment, we want to investigate the impact of using WVT on case without thin dielectric. Comparing the speedup data in Table I and Table II, it is obvious that the WVT is much useful for the case with thin dielectrics. Table II validates the efficiency of presented techniques again.

TABLE II
The Computational Times for Two Cases without Thin Dielectrics

	Time (s)		Speedup of G to O	Time (s)	Speedup of GW to O
	O	G		GW	
Case 3	3.62	2.01	1.8	0.21	17
Case 4	4.30	2.81	1.5	0.31	14

B. Exieriments of Parallel Computing

On the 8-core machine, the computational time of the parallel implementation of RWCap(GW) vs. the number of threads is shown in Table III. We run the first test case for 300,000 walks with different number of threads. Table III verifies the high efficiency of the parallel program. For other test cases, over 7X speedup is also achieved by RWCap on the 8-core machine, without loss of accuracy.

TABLE III
The Time of Parallel RWCap(GW) for Case 1

# Threads	Time (s)	Speedup	Efficiency
1	33.308	1.00X	100%
2	16.877	1.97X	99.7%
4	8.698	3.83X	96.8%
8	4.45	7.47X	93.4%

VI. Conclusions

An efficient FRW algorithm for multi-dielectric capacitance is presented. With the FDM, the Green's function and weight value for two-dielectric cube domain are numerically solved and stored as sets of GFTs and WVTs. Utilizing these tables for specified process technology, the FRW-based capacitance extraction can be accelerated by several tens to over a hundred times. The techniques of space management and parallel computing are also utilized; the former accelerates the FRW by at least several times for large structures, while the latter achieves 7.5X speedup on an 8-core machine. Numerical results on 45nm-technology interconnects and the comparison with FastCap [1] and CAPEM [8] validate the high efficiency, parallelism and accuracy of the proposed techniques.

References

[1] K. Nabors and J. White, "FastCap: A multipole accelerated 3-D capacitance extraction program," *IEEE Trans. CAD*, Vol. 10, pp. 1447–1459, Nov. 1991.

[2] Y. L. Le Coz and R. B. Iverson, "A stochastic algorithm for high speed capacitance extraction in integrated circuits," *Solid-State Electronics*, Vol. 35, No. 7, pp. 1005-1012, 1992.

[3] R. B. Iverson and Y. L. Le Coz, "A floating random-walk algorithm for extracting electrical capacitance," *Mathematics and Computers in Simulation*, Vol. 55, pp. 59-66, 2001.

[4] Y. L. Le Coz, H. J. Greub and R. B. Iverson, "Performance of random-walk capacitance extractors for IC interconnects: A numerical study," *Solid-State Electronics*, Vol. 42, No. 4, pp. 581-588, 1998.

[5] T. A. El-Moselhy, I. M. Elfadel, and L. Daniel, "A capacitance solver for incremental variation-aware extraction," in *Proc. ICCAD*, pp. 662-669, 2008.

[6] J. N. Jere and Y. L. Le Coz, "An improved floating-random-walk algorithm for solving the multi-dielectric Dirichlet problem," *IEEE Trans. MTT*, Vol. 41, No. 2, pp. 325-329, 1993.

[7] S. H. Batterywala, and M. P. Desai, "Variance reduction in Monte Carlo capacitance extraction," in *Proc. VLSI Design*, pp. 85-90, 2005

[8] M. P. Desai, "The Capacitance Extraction Tool," http://www.ee.iitb.ac.in/~microel/download.

[9] G. M. Royer, "A Monte Carlo procedure for potential theory problems," *IEEE Trans. MTT*, Vol. 19, No. 10, pp. 813-818, 1971.

[10] H. Samet, *Applications of Spatial Data Structure*, Addison-Wesley, Reading, MA. 1990.

[11] M. Matsumoto and T. Nishimura, "Mersenne twister: A 623-dimensionally equidistributed unifrom pseudo-random number generator," *ACM Trans. MCS*, Vol. 8, No. 1, pp. 3-30, Jan. 1998.

[12] D. Deschacht, S. de Rivaz, A. Farcy, T. Lacrevaz, and B. Flechet, "Keep on shrinking interconnect size: Is it still the best solution?" in *Proc. IEMT*, pp.1-4, Nov. 2010.

[13] W. Yu and Z. Wang, "Enhanced QMM-BEM solver for three-dimensional multiple-dielectric capacitance extraction within the finite domain," *IEEE Trans. MTT*, Vol. 52, No. 2, pp. 560-566, 2004.

[14] W. Yu, M. Zhang, and Z. Wang, "Efficient 3-D extraction of interconnect capacitance considering floating metal-fills with boundary element method," *IEEE Trans. CAD*, Vol. 25, No. 1, pp. 12-18, 2006.