An Efficient Quasi-Multiple Medium Algorithm for the Capacitance Extraction of Actual 3-D VLSI Interconnects

Wenjian Yu  
Department of Computer Science and Technology  
Tsinghua University  
Beijing 100084, China  
Tel: 8610-62785564  
Fax: 8610-62781489  
e-mail: yuwj@tiger.cs.tsinghua.edu.cn

Zeyi Wang  
Department of Computer Science and Technology  
Tsinghua University  
Beijing 100084, China  
Tel: 8610-62785564  
Fax: 8610-62781489  
e-mail: wangzy@tsinghua.edu.cn

Abstract—In this paper, the quasi-multiple medium (QMM) method based on the direct boundary element method (BEM) is used to extract the capacitance of three-dimensional (3-D) VLSI interconnects with multiple dielectrics. Each dielectric in 3-D VLSI parasitic capacitor is cut into several fictitious mediums, so it produces much fewer non-zero entries to the total coefficient matrix. As a result, the QMM algorithm can greatly reduce the CPU time and memory space used in the capacitance extraction. To characterize the efficiency of QMM algorithm, we discuss accuracy, storage and timing statistics in comparison with the analogous information from non-QMM algorithm.

I. Introduction

In VLSI circuits, with rapid increase of device density and working frequency, the electrical characteristics of interconnects are becoming more important factors governing the circuit performances such as delay, power consumption and reliability etc. This has increased the interest in efficient methods for calculating electrical parameters of interconnects.

Many numerical methods have been used to calculate the capacitance matrix of interconnects [1-10]. Generally they can be classified as the finite difference method (FDM) [5,7], finite element method (FEM) [6] and boundary element method (BEM) [1-4, 8-10]. Compared with the FDM and FEM, the main advantages of BEM are higher accuracy, less degree of freedom and stronger ability to handle complex 3-D structure [11,12].

Among the derivatives of boundary element method, the direct BEM is more convenient for treating the problems with the mixed Neumann and Dirichlet boundary conditions because there are two variables of potential and its normal derivative in the direct boundary integral equation [12,13]. So, the direct BEM is more suitable for simulation of the parasitic capacitors with finite domain and multiple dielectrics. However, the BEM generally leads to non-symmetric matrix equations and the matrix for each homogeneous region is dense. This causes a great deal of time and memory consumption in forming and solving the system of linear equations [11].

Recently, a new quasi-multiple medium (QMM) method based on the direct BEM was proposed to accelerate the capacitance extraction [14]. It utilizes the localization character of the direct BEM to transfer the coefficient matrix into a very sparse block matrix. Using the technology of storing sparse matrices and the iterative equation solver, the QMM method will greatly accelerate the computation of the direct BEM. A primary computational complexity analysis of the QMM method was given in [14] and it was concluded with a nearly linear relationship of N, where N is the number of the boundary elements. In [14], only a simplified problem was analyzed, which had a single dielectric with some metal cubes in it, however, many aspects in processing actual 3-D VLSI interconnects had not been considered. In this paper, first we give a theoretic analysis of the QMM algorithm for the complex geometry of actual 3-D interconnects capacitor, then we present an implementation of the QMM algorithm and discuss some important aspects of it in detail. Numerical results show that the QMM method can drastically reduce the calculation time and the memory space used in the actual 3-D interconnects capacitance extraction.

II. Direct BEM Computation for 3-D Interconnect Capacitance

The parasitic capacitance can be obtained through the computation of the normal electrical field intensity on the Dirichlet boundary (surfaces of conductors) [3]. In three-dimensional multiple dielectrics domain $\Omega = \cup_{i} \Omega_{i}$, electrical potential $u$ fulfill the following Laplace equations and mixed boundary conditions.

$$\left\{ \begin{array}{ll} \nabla^{2} u = 0 & \text{In } \Omega_{i} (i = 1, \ldots, M) \\ u = u_{0} & \text{On } \Gamma_{u}^{n} \\ \frac{\partial u}{\partial n} = q_{0} = 0 & \text{On } \Gamma_{q} \end{array} \right.,$$

(1)

where $\Omega_{i}$ is 3-D domain of the $i$th dielectric. $\Gamma_{u}$ is the Dirichlet boundary (surfaces of conductors), where the electrical potential $u$ is known and determined by the bias voltage; $\Gamma_{q}$ is the Neumann boundary (surfaces of dielectrics), where the normal electrical field intensity $q$ is supposed to be zero. $n$ stands for the unit normal vector of the boundary. $\varepsilon_{e}$ is permittivity of the $i$th dielectric, and $M$ is the number of dielectrics in the domain $\Omega$.

Besides, there is compatibility of $u$ and $q$ along the interface between two adjacent dielectrics $a$ and $b$:

$$\left\{ \begin{array}{ll} \varepsilon_{e} \cdot \frac{\partial u_{a}}{\partial n_{a}} = -\varepsilon_{b} \cdot \frac{\partial u_{b}}{\partial n_{b}}, \\
u_{a} = u_{b} \end{array} \right.,$$

(2)

With the fundamental solution $u^{*}$ as weighting function, the Laplace equations in (1) for one setting of the bias voltage...
are transformed into following direct boundary integral equations (BIEs) by using the Green’s formula [11,12]:
\[ c \cdot u_i + \int_{\partial \Omega} q^i u d\Gamma = \int_{\partial \Omega} u^i q d\Gamma, \quad (i=1,\ldots,M) \]  
(3)
where \( u_i \) is the electrical potential at the source point \( s \), \( c \) is a constant dependent on boundary geometry near the point \( s \), \( q^i \) is the \( u^i \)’s derivative along the outward normal direction of boundary \( \partial \Omega \), and \( \partial \Omega \) is the boundary surrounding dielectric \( i \).

Employing constant elements, we obtain discretized BIEs. By evaluating these expressions at a set of locations, one can generate a matrix system of equations for each dielectric [11].

\[ H^i \cdot u^i = G^i \cdot q^i, \quad (i=1,\ldots,M) \]  
(4)
where \( u^i \) is the column vector of electrical potential, \( q^i \) is the column vector of normal electrical field intensity, and \( H^i \) and \( G^i \) are corresponding coefficient matrices, respectively. Both vectors of \( u^i \) and \( q^i \) have degree of \( N_i \).

Matrix equations (4) can be put together by employing the compatibility equations (2). Then after putting all unknowns to the left side of equal mark and making corresponding movement of corresponding columns in \( H^i \) and \( G^i \), we obtained a system of linear equations [3].

\[ Ax = f \]  
(5)

After solution of equation (5) by using the generalized minimal residual (GMRES) algorithm, the parasitic capacitance can be evaluated from the normal electrical field intensity on the Dirichlet boundary.

III. The Principle of QMM

For a problem with multiple media, there are two kinds of interaction among the discrete boundary variables, direct interaction and indirect interaction. If a group of variables are simultaneously involved in the same discrete BIE (3), there exist the direct interaction among them, otherwise the indirect interaction. That implies the direct interaction is limited to the boundary variables that belong to the same dielectric. We call this the localization character of direct BEM.

In the linear system (5), coefficient matrix \( A \) reflects the situation of interaction among the discrete boundary variables. If there is the direct interaction between two variables, they form a non-zero entry in the coefficient matrix \( A \), otherwise a zero entry. For a problem with multiple dielectrics, the localization of direct BEM make matrix \( A \) become one with sparse blocks, from which we can benefit in solving the system of algebraic equations (5). In Fig. 1, we show a typical capacitor with two dielectrics and the population of corresponding matrix \( A \) generated by the direct BEM.

The QMM method takes full advantage of the localization of direct BEM. For a single medium, the direct BEM generates a dense and non-symmetric matrix. But, we can suppose that one dielectric with the permittivity \( \varepsilon \) consists of \( Q \) blocks of quasi-multiple medium, whose permittivities are all the same as the original permittivity \( \varepsilon \). And the fictitious mediums do not overlap each other, as shown in Fig. 2. Thus, the problem with single medium is transferred into a problem with multiple fictitious mediums, and can be solved by the direct BEM for multiple mediums. We call this quasi-multiple medium (QMM) method.

![Fig. 2. A single dielectric capacitor with Cartesian coordinate system is cut into \( Q=Q_1, Q_2, \) fictitious mediums.](image)

With a moderate value of parameter \( Q \), the coefficient matrix \( A \) generated by the QMM method will be one with much sparsity because of the localization of the direct BEM. Using the storage techniques of sparse blocked matrix and the iterative solution approaches such as GMRES, the QMM method will greatly reduce the computation time and the memory usage.

It should be pointed out that the QMM method adds some unknowns to the overall problem, which are introduced by the additional fictitious interfaces of quasi-multiple medium blocks. This effect tends to make the computation require more computer resources. But as we can see later, in most cases the advantage taken from the sparsity of the coefficient matrix might greatly outweigh the disadvantage of adding some unknowns.

IV. Application of QMM Method in Actual Interconnect Capacitance Extraction

An interconnect capacitor cut from a real design usually has stratified structures, and has many conductors embedded

![Fig. 1. (a) A two-dimensional problem with two dielectrics, and (b) the corresponding sparse blocked matrix \( A \).](image)
in multiple dielectrics. Coupling capacitance between conductors on critical path (master conductors, bias voltage is set to 1v) and other conductors within the simulating region (environment conductors, bias voltage is set to 0v) is to be computed.

In order to apply the QMM method in actual 3-D interconnect capacitance extraction, we firstly need to cut the original dielectric layers into some fictitious medium blocks. Then we can use the direct BEM to calculate the capacitance with the new multi-dielectric structure. We expect to improve the computational efficiency by using the QMM method.

A. Theoretical Analysis

Here we analyze the CPU time used by the QMM algorithm, and compare it with the non-QMM algorithm. The total CPU time used in the 3-D interconnect capacitance extraction with the direct BEM

\[ t = t_{\text{gen}} + t_{\text{sol}} + t_{\text{aux}}, \]  

(6)

where \( t_{\text{gen}} \) is the time spent in generating the equation (5), \( t_{\text{sol}} \) is the time spent in solving (5), and \( t_{\text{aux}} \) stands for the time spent in other supplementary procedures, including input of structure data and partition of boundary elements. Generally speaking, the sum of \( t_{\text{sol}} \) and \( t_{\text{gen}} \) accounts for more than 90% of the total CPU time. Using the storage technology of sparse matrix and the iterative equation solvers such as GMRES, we can then get the following expressions:

\[ t_{\text{gen}} \propto Z, \quad t_{\text{sol}} \propto Z \cdot k, \]

(7)

where \( Z \) stands for the number of non-zero entries of matrix \( A \), and \( k \) stands for the number of iterations.

If we ignore \( t_{\text{aux}} \), and suppose that there is not much difference between non-QMM algorithm and QMM algorithm in the number of iterations, the speed up ratio of QMM algorithm

\[ R_{\text{speed-up}} = t / t' = Z / Z', \]

(8)

where \( t \) and \( t' \) are the CPU time of non-QMM algorithm and QMM algorithm respectively, and \( Z \) and \( Z' \) stand for the number of non-zero entries of matrix \( A \) generated by non-QMM algorithm and QMM algorithm respectively. This expression shows the ratio of number of non-zero entries in matrix \( A \) reflects the speed up ratio of the QMM algorithm.

In non-QMM computation, the \( i \)th dielectric layer has the following contribution to the total number of non-zero entries in matrix \( A \):

\[ Z_i = (a + b)(a + 2b) \]

(9)

where \( a \) stands for the number of first class boundary elements (on conductors surfaces and dielectrics surfaces, on which there is only one variable), and \( b \) stands for the number of second class boundary elements (on interfaces of dielectrics, on which there are two variables). While in QMM computation with \( Q \) quasi-multiple medium blocks, we have the similar formulation:

\[ Z'_i = \sum_{j=1}^{Q} (a_j + b_j)(a_j + 2b_j), \]

(10)

where \( a_j \) and \( b_j \) stand for the number of first and second class boundary elements in the \( j \)th fictitious medium block respectively. Suppose in QMM algorithm, the partitions of boundary elements on the actual boundaries are the same with that of non-QMM algorithm, the following relationship can be obtained.

\[ \sum_{j=1}^{Q} a_j = a, \quad \sum_{j=1}^{Q} b_j = b + 2c \]

(11)

where \( c \) is the number of boundary elements on the additional interfaces of fictitious medium blocks. Combining the formulation (10) and (11), we can get the optimal \( Z'_i \),

\[ Z'_{i, \text{min}} = \frac{Z_i + 2c(3a + 4b) + 8c^2}{Q} \]

(12)

So, the optimal speed up ratio achieved by the QMM algorithm for the \( i \)th dielectric layer

\[ R_i = Z_i / Z'_{i, \text{min}} = \frac{Q}{Z_i + 2c(3a + 4b) + 8c^2}. \]

(13)

And we can obtain the total speed up ratio by calculate the speed up ratios for each layer.

\[ R = \frac{Z}{Z'} = \frac{\sum Z_i}{\sum Z'_i} = \left( \frac{\sum Z'_i}{\sum Z'_i} \right) R_i. \]

(14)

That means the total speed up ratio is a linear combination of ratio for each layer.

From the above analysis, we can get some points of our QMM algorithm. 1), the QMM algorithm cuts every dielectric layer at first, then partition boundary elements and perform following computation. So, complex strategy of cutting will not be adopted because it costs much CPU time. 2), If \( Q \) is very large, there exist many fictitious medium blocks in the capacitor. This not only introduces a great deal of additional elements on the interfaces among fictitious medium blocks, but also make the original surfaces become many pieces. Thus, the discretized boundary elements on these pieces of surfaces may be much more than that of the assumption (11). So increasing \( Q \) unlimited will not improve the computational efficiency. 3), From the equations (13) and (14), we can find out that the expected efficiency of QMM algorithm will be better for the interconnect capacitor with more conductors (\( Z \) is very large). 4), The element partition on the interfaces among fictitious medium blocks is very important. It not only affects the computational accuracy, but also influences the key item \( c \) in the QMM speed up ratio formulation (13).

B. Algorithm Description

The major steps of the QMM algorithm are as follows:

1) Read in the data describing a 3-D interconnect capacitor.
2) Set the element-partitioning gaps for each boundary surface.
3) For \( i = 1 \) to \( M \)
   Perform QMM cutting to the \( i \)th dielectric;
   For \( j = 1 \) to \( \text{ConductorNumberInLayer}[i] \)
     If (the \( j \)th conductor intersect additional interfaces of fictitious mediums)
     Cut the conductor \( j \) and describe each part of it as a single conductor block;
     Set the containing relationships of the conductor
blocks and the fictitious medium blocks;

EndIf
EndFor
EndFor

4) Organize the medium blocks and conductors blocks into new object lists.
5) Set the element-partitioning gap for each boundary surface of the new capacitor structure.
6) Perform numerical integral and form the equation (5).
7) Solve the equation (5) with preconditioning GMRES and output the capacitance result.

C. Cutting of Dielectrics and Conductors

In order to decrease the additional processes brought by the QMM algorithm, we adopt a simple cutting strategy. Since every cutting strategy corresponding to the formula (12) is unavailable. So we adopt a strategy of proportional-spacing cutting. That is, \( m \)-1 fictitious planes perpendicular to X-axis cut the dielectric equally along the X-axis, and \( n \)-1 fictitious planes perpendicular to Y-axis cut the dielectric equally along the Y-axis. Our test reveals that, manually changing the positions of cutting planes may only bring 10% more benefit than the proportional-spacing cutting at most.

So, every dielectric layer is decomposed into \( Q=mn \) fictitious medium blocks. If \( Q \) is very little or very large, the QMM algorithm can not give the best performance for speeding up computation, neither. Therefore, we let \( m \) and \( n \) be a moderate value respectively. Since no rule to choose the optimal value of \( m \) and \( n \), we firstly use a trial-and-error process to decide them for each specific structure, and then we get an empirical formula for a great deal of examples.

D. Partition of Boundary Element

In application of BEM, the partition of boundary element is very important. It affects both speed and accuracy of BEM computation. In this paper, we adopt a strategy of non-uniform density partitioning. Thus we can partition the boundaries into less elements without loss of accuracy.

There are two kinds of boundary. One can be treated as surface without holes, and the other should be treated as surface with some polygon holes. Using the scan line algorithm, those surfaces with holes can be further treated as composition of smaller trapezoid [15]. Hence, both kinds of boundary surface consist of the trapezoids, which are called mother elements and need to be further divided into boundary elements.

According to electrostatic analysis, the electrical field intensity on the boundary surface of conductor, especially the master conductor, generally is largest in the simulated region. Besides, the electrical field intensity at boundary points near the master conductor is large too. So, for each mother element to be partitioned, the mesh number along the two directions should be determined by its type, position, and size etc. The larger is the electrical field intensity at a mother element, the more densely should it be partitioned.

For the additional fictitious surfaces introduced by QMM, we also use different partition density according to the above electrostatic analysis. For each dielectric layer, the partition density of fictitious surfaces is different. In the dielectric layer containing master conductor, fictitious surface is partitioned more densely. While in the dielectric layer far from the master layer, the partition density can be much less.

In QMM algorithm, the interface of dielectric layer is cut into small pieces, and some fictitious surfaces (which may be surface with holes) are introduced. So the partition of boundary element in QMM is more complex than that of non-QMM(Fig.4 shows a partition of bottom surface of the master dielectric layer in capacitor shown in Fig.3). This complexity makes it very hard to give more precise analysis of the QMM algorithm.

E. Storage of Sparse Matrix

In the QMM algorithm, there are much more medium blocks than in non-QMM algorithm. So, the non-zero blocks in coefficient matrix are much more, they distribute in the global matrix dispersedly. If we store these coefficient blocks in different arrays as in common storing technique, the switching among arrays will cost much CPU time in every

![Fig. 3. A typical 3-D interconnect capacitor with 5 dielectrics which are cut into 3×2 structures](image)

![Fig. 4. The boundary element partition of one dielectric interface in the capacitor shown in Fig. 3](image)
matrix-vector multiplication performed in GMRES algorithm, and it decreases the efficiency of equation solving greatly. So in this paper, we use a special structure of two-dimensional array to refine the storing of matrix A (Fig. 5). Therefore, all non-zero matrix entries are stored in a two-dimensional array, the operation of matrix-vector multiplication is simplified. Experimental results reveal that the refined storing structure improves the speed of matrix-vector multiplication in QMM computation significantly.

![Image](image_url)

Fig. 5. The two-dimensional array structure to store matrix A.

V. Numerical Results and Conclusion

This section presents some examples to demonstrate the efficiency of the QMM algorithm. Here we use Sun Ultra E450 to carry out our experiments. In all data tables the unit of time is second, and the unit of capacitance is $10^{18}$ farad. The unit of memory is megabyte.

A. Three-Layer $k$-$k$ Crossovers

The $k$-$k$ ($k$=2, 4, 6) bus crossing conductors immersed in three dielectric layers with a ground plane at the very bottom of the structure is calculated using QMM and non-QMM algorithm. Fig. 6 shows the structure of 6x6 crossover. The structure parameters are as follows. The height of each dielectric layer is 5 μm. Each metal line is a 3x3x28 cube (unit in μm). The space between two adjacent metal lines on the same layer is 2 μm. The distance between the outmost metal line and the natural boundary of dielectric is 1 μm. Every metal line touches the bottom of the dielectric which embed it. The relative permittivities are all chosen to be 1.0 for the sake of simplicity. Removal of conductor 1, 6, 7 and 12 produce the 4x4 crossover. And removal of conductor 2, 5, 8 and 11 again produce the 2x2 case. The total capacitance to be computed is that between master conductor 3 and the other conductors. The cut number along X and Y directions in the QMM algorithm are set to 6, 6, and 4 for the three $k$-$k$ ($k$=6, 4, 2) crossover respectively.

Table I shows the numbers of non-zero entries in matrix A and the iteration step in the two algorithms. From it, we can see that the QMM generates much fewer non-zero coefficients than the non-QMM, so high speed up ratio can be expected. Table I also shows that iteration step increase in the QMM while using the same GMRES solver.

![Table](table_url)

Table II shows the total capacitance, CPU time and memory usage of the QMM compared with those of non-QMM. The QMM is about 4-10 times faster than non-QMM with the difference in the results less than 3%. And the QMM uses about 2-4 times of less memory usage than non-QMM. The increase of iteration number shown in Table I could account for the difference between the actual speed up ratio and the expected one.

B. 3-D Structures Cut from a Real Design

We have also compared the QMM results with the non-QMM on three large 3-D examples using five metal layer technology. These examples all have metals distributing from layer 3 to layer 4, and include many crossovers. The first example has 26 metal lines, while the second and the third have 31 and 34 metal lines respectively. The cut number along X and Y directions in the QMM are different for the three examples. They are 3x6, 3x8 and 3x7 respectively. The results are shown in Table III.

As shown in Table III, the QMM is about eight times faster than the non-QMM while preserving high accuracy. It also can be found that the QMM uses about 1/4 to 1/3 of memory that the non-QMM uses.

Above two experiments reveal that the QMM significantly reduce the CPU time and memory space while preserving accuracy. The QMM algorithm shows great efficiency in the capacitance extraction of actual 3-D interconnects. The following two aspects also should be pointed out. 1) The speed up ratio is relevant to the geometry of the parasitic capacitor being simulated. Higher speed up can be obtained while computing parasitic capacitors containing larger or more conductors. 2) A more effective cutting strategy of QMM would be considered in the future research work.

VI. Acknowledgements

This work is partly supported by “National Foundation
for Key Basic Research” G1998030404 and Synopsys Inc.,
and also belongs to project 69876024 supported by “National
Natural Science Foundation of China”. The authors would
like to thank for their financial supports.

TABLE II
Comparison of CPU Time and Capacitance for k×k Crossover Problem

<table>
<thead>
<tr>
<th>Ele-Num</th>
<th>Memory</th>
<th>Capac.</th>
<th>Time</th>
<th>Ele-Num</th>
<th>Memory</th>
<th>Capac.</th>
<th>Error (%)</th>
<th>Time</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>6×6</td>
<td>5547</td>
<td>62</td>
<td>1699</td>
<td>161.4</td>
<td>6822</td>
<td>17</td>
<td>1738</td>
<td>2.3</td>
<td>16.6</td>
</tr>
<tr>
<td>4×4</td>
<td>4475</td>
<td>42</td>
<td>1636</td>
<td>104.2</td>
<td>5630</td>
<td>15</td>
<td>1673</td>
<td>2.3</td>
<td>14.6</td>
</tr>
<tr>
<td>2×2</td>
<td>3238</td>
<td>25</td>
<td>1157</td>
<td>56.1</td>
<td>3644</td>
<td>11</td>
<td>1180</td>
<td>2.0</td>
<td>13.4</td>
</tr>
</tbody>
</table>

TABLE III
Comparison of QMM Algorithm and Non-QMM Algorithm for Three Real Problems

<table>
<thead>
<tr>
<th>Ele-Num</th>
<th>Iteration</th>
<th>Memory</th>
<th>Capac.</th>
<th>Time</th>
<th>Ele-Num</th>
<th>Iteration</th>
<th>Memory</th>
<th>Capac.</th>
<th>Error (%)</th>
<th>Time</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5057</td>
<td>40</td>
<td>48</td>
<td>20760</td>
<td>139.3</td>
<td>5772</td>
<td>42</td>
<td>14</td>
<td>20724</td>
<td>-0.2</td>
<td>19.2</td>
</tr>
<tr>
<td>2</td>
<td>6214</td>
<td>41</td>
<td>76</td>
<td>24359</td>
<td>236.0</td>
<td>7234</td>
<td>47</td>
<td>19</td>
<td>24148</td>
<td>-0.9</td>
<td>29.2</td>
</tr>
<tr>
<td>3</td>
<td>5451</td>
<td>38</td>
<td>58</td>
<td>19603</td>
<td>172.0</td>
<td>6318</td>
<td>40</td>
<td>17</td>
<td>19681</td>
<td>0.4</td>
<td>23.0</td>
</tr>
</tbody>
</table>

References


