ICE: Incremental 3-Dimensional Capacitance and Resistance Extraction for an Iterative Design Environment *

Yanhong Yuan Prithviraj Banerjee

Department of Electrical and Computer Engineering
Northwestern University, Evanston, IL 60208
E-mail: {yuanyh,banerjee}@ece.nwu.edu

Abstract

In this paper, we discuss the 3-Dimensional(3-D) capacitance and resistance extraction within an iterative design environment, where small changes are made to the 3-D structures. We present a bounded incremental algorithm for accurate and fast 3-D extraction in such a design environment, based on the Boundary Element Method(BEM). The incremental algorithm can re-utilize the computation results of previous extractions and rapidly re-compute the new parasitic parameters in response to the design changes made to the layout. The incremental algorithm has been implemented in the ICE tool. Experimental results on a set of 3-D interconnect structures show that the incremental algorithm is efficient for the iterative design methodology. For one large structure, the incremental extraction is over 20 times faster than the full extraction without using the incremental algorithm. To the best of our knowledge, this is the first reported work on an incremental algorithm for capacitance and resistance extraction.

1. Introduction

Accurate and fast extraction of the capacitance(C) and resistance(R) in complicated 3-dimensional(3-D) interconnects is important for VLSI system design. Many extraction approaches have been published [1, 2, 3].

This paper addresses the 3-D extraction in an *iterative design environment*. Figure 1 shows the typical methodology for such an iterative design approach. An iterative design approach exhibits the feature of *incremental improvement*. As shown in Figure 1, designers make small changes to a few nets of the layout based on a previous timing simulation run. In response to those changes, it is necessary to *recompute* the resistances and capacitances of only those nets that are affected by those design changes.

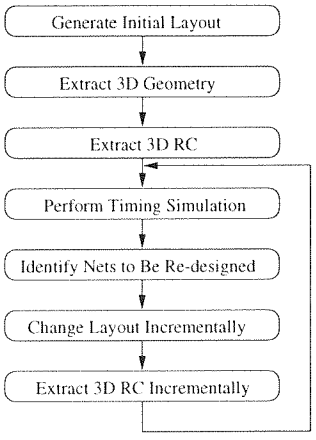

Figure 1. Methodology for an iterative layout design.

One method to handle the parasitic extraction in an iterative design is to use a parasitic database and pattern matching. As circuit density and complexity continues to increase, the interconnect structures are becoming increasingly complicated and it has been more and more difficult to predefine patterns for matching.

In this paper, we propose to develop a method for extracting the capacitances and resistances directly from integral equation solutions during an iterative layout design. We present an algorithm for *incremental extraction* of 3-D capacitance and resistance in such a design environment. It can effectively re-utilize the computation results of a previous extraction step and rapidly re-compute the new parasitic parameters based on the design changes made to the layout. Because complicated details of the circuits can be modeled, the algorithm is more accurate than the *database* based approach for the very deep sub-micron design.

We develop the incremental algorithm based on the

*This research was supported in part by the Advanced Research Projects Agency under contract DAA-H04-94-G0273 administered by the Army Research Office.

0-7695-0104-4/99 $10.00 © 1999 IEEE

Boundary Element Method(BEM) which is called the *direct BEM method* in the fundamental book by Brebbia[4]. This BEM model can be used to calculate both the capacitance and resistance with the same procedure [5, 6]. The incremental algorithm has been implemented in the ICE tool: Incremental 3-D Capacitance and resistance Extraction for an iterative design environment. Experimental results on a set of 3-D interconnect structures show that the incremental algorithm is efficient for the iterative design.

Incremental algorithms have been proposed in the past in a variety of VLSI CAD tools, such as the event driven logic simulation, design rule check(DRC) in MAGIC, and rip-up and reroute detailed routers. To the best of our knowledge, this is the first reported work on an incremental algorithm for 3D capacitance and resistance extraction.

2. Problem formulation

We briefly review the direct BEM method[4] and its application to parameter extraction. A 3-D Laplace equation and associated boundary conditions are solved to precisely compute the capacitance or resistance. Laplace equation and its mixed boundary conditions for one setting of the bias voltages can be transformed into the following *direct BEM integral equation*:

$$c_s u_s + \int_\Gamma q^* \, u \, d\Gamma = \int_\Gamma u^* \, q \, d\Gamma, \qquad (1)$$

where u_s is the potential at an source point $S(x_s, y_s, z_s)$, and c_s is a constant depending on the geometry of the boundary Γ near the point S. In a 3-D situation, the fundamental solution u^* is $\frac{1}{4\pi r}$. r is the distance between the source point S and the field point (x, y, z). Boundary Γ consists of conductor surfaces, dielectric interfaces and the Neumann surface.

Γ can be divided into a series of boundary elements. By applying the interpolation functions, (1) can be re-written as:

$$\sum_j^n H_{ij} u_j \quad = \quad \sum_j^n G_{ij} q_j, \quad i = 1, ..., n \qquad (2)$$

where n is the total number of nodes on Γ, and $H_{ij} = \sum_k (\int_{\Gamma_{jk}} q^* \, \Phi_{jk} \, d\Gamma)$, $G_{ij} = \sum_k (\int_{\Gamma_{jk}} u^* \, \Phi_{jk} \, d\Gamma)$. Φ_{jk} is the interpolation function of node j in element Γ_{jk}. Substituting boundary conditions into (2), a linear equation system can be obtained as:

$$Ax \quad = \quad f, \qquad (3)$$

where the column vector x has n unknown variables.

Solving this linear system, we can *directly get the normal electric field* q on the Dirichlet boundary, and finally, we can calculate the capacitance(C) or resistance(R) as:

$$C = \int_{\Gamma_u} \epsilon \, q \, d\Gamma, \quad R = \frac{1}{\int_{\Gamma_u} \rho \, q \, d\Gamma}, \qquad (4)$$

where ϵ is the permittivity, and ρ is the resistivity.

3. Incremental algorithms in ICE

ICE takes as inputs the incremental design changes, and performs the incremental 3-D capacitance and resistance extraction on the modified layout structure, based on a previous extraction. The ICE tool consists of the following steps:

1. *Perform initial R&C extraction on the initial layout design, using the direct BEM method.*
2. *Iterative incremental R&C extraction*
   (a) *Perform a sequence of design changes to the identified nets based on the result of timing simulation;*
   (b) *Regenerate the mesh according to the specified operations in an incremental manner;*
   (c) *Reconfigure the matrix A rapidly according to the operations in an incremental manner;*
   (d) *Incrementally solve the new linear system rapidly to obtain the new R&C values, using the solution of previous extraction as an initial guess for the new solution;*
   (e) *If further design changes should be made according to the timing simulation based on the new R&C values, goto (a);*
   else done;

Currently, four basic operations are defined in ICE, and they can be used to perform the design changes commonly required in an iterative design environment, such as moving a conductor around, making a contact(or via), making a connection(bent) and so on:

MOVE: Move a conductor;
ADD: Insert a conductor;
DELETE: Delete a conductor;
STRETCH: Resize a conductor;

In the context of this paper, we will describe two main steps in the algorithm: *incremental system re-configuration* and *incremental re-solution*.

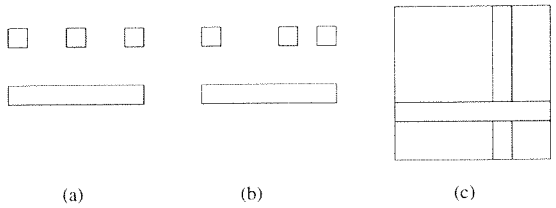

Figure 2. An incremental MOVE operation: (a) The original structure; (b) Moving the shaded conductor; (c) The part of the matrix that need to be recalculated(shaded area).

3.1. Incremental re-configuration

For a clear illustration, look at the standard configuration of matrix A in equation (3). Each node and each element is related to some particular entry in the matrix. There are a group of nodes and elements incident on the surface of a conductor. So, a conductor will be related to some block of entries in the matrix. For example, in Figure 2, the shaded part in (c) is related to the moved conductor(shaded). Therefore, changing the design of some conductor results in modifying the related part in the system matrix A. When the changed part is relatively small, the new system matrix reflecting the new design can be re-configured rapidly by performing some local re-calculation of the matrix.

The incremental re-configuration for other operations is handled in the similar way. To reduce the size of the problem and improve the accuracy, an *adaptive mesh* is employed with finer discretization at or near edges, contacts and corners. So, when a design is changed, an adaptive mesh will be re-configured locally to reflect the expected rapid variation in electric field and potential in these regions.

3.2. Incremental solution

To avoid the $O(n^3)$ cost of solving (3) with Gaussian elimination, a conjugate residual like algorithm GMRES[7] is commonly used. An initial guess x_0 to the linear system is required. An all-zero vector is usually used. In an incremental design, when the design change is small, it is reasonable that the solution will not change significantly. Therefore, in the incremental algorithm, instead of using an all-zero vector as x_0 each time, the solution of the previous extraction is used as x_0 for the modified linear system. This approach results in significantly fast convergence to the final solution in much less number of iteration steps.

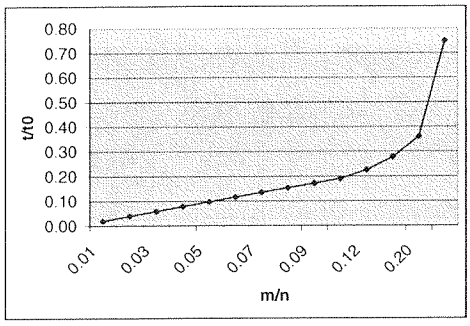

Figure 3. Ratio t/t_0 as a function of m/n.

3.3. Running time

Assume m is the part of the structure that is affected by an operation, and n is the size of the *resulting structure* after the change, both in terms of the number of elements. For DELETE operation, we use *invalidation* for the corresponding matrix entries, so the computation time is $O(1)$. The time of the system re-configuration for operations ADD, MOVE and STRETCH is dependent on the size of changes made to the layout. An upper bound can be obtained by examining the standard matrix configuration. In that case, the time to incrementally reconfigure the matrix for MOVE, ADD or STRETCH operation is $t = O(2mn - mm)$. In contrast, the time to construct the full matrix for the resulting design without incremental computation is $t_0 = O(n^2)$. The ratio t/t_0 as a function of m/n is displayed in Figure 3, where m/n reflects the portion of changes made to the structure for a particular operation. We see that when $m/n < 1/10$, which means that small changes are made, the curve changes all most linearly. Because the time it takes to re-extract the parasitic parameters depends only on the size of the change made to the design, the incremental algorithm is bounded[8].

4. Experimental results

The incremental algorithm has been implemented in an iterative extraction tool called ICE, which was written in the C programming language for execution on HP C-110 workstation with a 120MHz PA-7200 CPU and 128MB memory. A set of interconnect structures found in typical multi-layer VLSI process are used to test the algorithm. Table 1 lists some important characteristics of the test structures.

Table 2 summarizes the performance of the basic operations on the set of test structures. The *original time* T_{orig} gives the CPU time (seconds) to extract the *resulting struc-*

Table 1. Characteristics of test structures.

structure	small	medium	large
#metal layers	2	3	4
#conductors	6	45	100
#elements	1088	1872	2500

ture after performing some design change, using the extraction method *without incremental computation*. The *incremental time* T_{incre} is the time to *incrementally re-extract* the changed structure based on a previous extraction. Also shown in this column in the parenthesis are the numbers of iterations required for the convergence by using two different choices for the initial solution x_0. The first one is the number of iterations when an all zero vector is used as x_0. The second one is the number when the previous solution is used as x_0. It can be found that this results in extremely fast convergence to the new final solution in much less number of iterations. The "changed" column gives the amount of changes (in percentage) made to the structure. The last column presents the efficiency of the incremental algorithm.

We find that for the medium size structure, it takes less than 10% of the original time T_{orig} to incrementally re-extract the changed structure. For the large structure, the incremental extraction is about 25 to 50 times faster than the extraction without using the incremental algorithm.

For the large structure containing about 100 conductors, we perform a sequence of operations to modify the structure, i.e. {ADD, MOVE, DELETE, STRETCH}. The CPU times are reported in Table 3. The changed portion of the structure for each individual operation is roughly 1%. The time to extract the resulting structure is about 1300 seconds, by using the extraction method without incremental computation. The total time of the incremental re-extraction is about 95 seconds, which is 7% of the extraction time without using the incremental method.

Table 2. Times(s) of basic operations. The first number in the parenthesis is the number of iterations required for the convergence by using all zero vector as x_0. The second is the number when the previous solution is used as x_0.

structure	T_{orig}	T_{incre}	changed	$\frac{T_{incre}}{T_{orig}}$
		MOVE		
small	284	86(20/5)	16%	30%
medium	875	65(21/6)	3%	7.4%
large	1178	40(38/6)	1%	3.4%
		DELETE		
small	197	9(20/5)	15%	4.0%
medium	810	11(21/6)	3%	1.4%
large	1147	18(37/5)	1%	1.6%
		ADD		
small	284	91(20/10)	16%	32%
medium	875	73(21/8)	3%	8.3%
large	1178	59(38/16)	1%	5.0%
		STRETCH		
small	284	50(20/6)	17%	17%
medium	875	40(21/7)	2%	4.6%
large	1178	32(38/10)	0.5%	2.7%

Table 3. A sequence of operations.

ADD	MOVE	DELETE	STRETCH	GMRES
22	20	3	10	40

5. Discussions

We have presented an *incremental* capacitance and resistance extraction algorithm. Experimental results show that it is very efficient for an *iterative design environment*. We are presently studying new techniques which will reduce the turn-around time and memory usage of each iteration step.

References

[1] K. Nabors and J. White. "FASTCAP: A Multipole-Accelerated 3-D Capacitance Extraction Program". *IEEE Trans. on CAD*, pages 1447–1459, 1991.

[2] S. Kapur and D.E. Long. "IES^3: A Fast Integral Equation Solver for Efficient 3-Dimensional Extraction". In *Proc. ICCAD*, pages 448–455, 1997.

[3] W. Shi, J. Liu, N. Kakani, and T. Yu. "A Fast Hierarchical Algorithm for 3-D Capacitance Extraction". In *Proc. DAC*, 1998.

[4] C. A. Brebbia. **The Boundary Element Method for Engineers**. Pentech Press, London, 1978.

[5] S. Fukda, N. Shigyo, K. Kato, and S. Nakamura. "A ULSI 2-D Capacitance Simulator for Complex Structures Based on Actual Processes". *IEEE Trans. on CAD*, 39(9), 1990.

[6] Z. Wang and Q. Wu. "A Two Dimensional Resistance Simulator Using the Boundary Element Method". *IEEE Trans. on CAD*, pages 946–954, 1992.

[7] Y.Saad. **Iterative Methods for Sparse Linear Systems**. PWS Publishing Company, 1996.

[8] G. Ramalingam. **Lecture Notes in Computer Science:Bounded Incremental Computation**. Springer, 1991.