Using Articulation Nodes to Improve the Efficiency of Finite-Element based Resistance Extraction

A.J. van Genderen and N.P. van der Meijs
Delft University of Technology
Department of Electrical Engineering
Mekelweg 4, 2628 CD Delft, The Netherlands
email: arjan@cas.et.tudelft.nl

Abstract

In this paper, we describe how we have improved the efficiency of a finite-element method for interconnect resistance extraction by introducing articulation nodes in the finite-element mesh. The articulation nodes are found by detecting equipotential regions and lines in the interconnects. Without generating inaccuracies, these articulation nodes split the finite-element mesh into small pieces that can be solved independently. The method has been implemented in the layout-to-circuit extractor Space. All interconnect resistances of a circuit containing 63,000 transistors are extracted on an HP 9000/735 workstation in approximately 70 minutes.

1 Introduction

In deep submicron integrated circuits, the influence of interconnect resistances is becoming more and more important. This is caused by the fact that, due to the decrease in the width of the wires, the resistance per unit length increases. At the same time, due to the increase in chip size, the average length of the wires increases.

Significant values for interconnect resistances in integrated circuits may cause malfunctioning of the circuit, e.g. because of too large delay time values or because of too great a drop in voltage along supply lines. Therefore, it is important to compute the values of the interconnect resistances from the layout of the circuit, so that the behavior of the circuit can be verified before it is fabricated.

Common approaches to the extraction of VLSI interconnect resistances include the Finite-Element Method (FEM) [1, 2, 3] and the Finite-Difference Method (FDM) [4]. These methods solve the resistance extraction problem by discretizing the governing differential equation (i.e. the Laplace equation) and solving the resulting set of algebraic equations. This set of equations is sparse, symmetric and positive definite, and is usually solved by Gaussian elimination [2, 3, 5].

When compared to other methods for resistance extraction, such as Polygonal Decomposition (as in [6]), Conformal Transformation (as in [7]) and the Boundary-Element Method (BEM) (as in [8]), the advantages of the FEM include general applicability, robustness, good accuracy and the possibility of accurately extracting RC models [2, 9]. Disadvantages, however, are those of longer computation times and higher memory requirements.

More specifically, the computation time for solving the set of finite-element equations increases more than linearly with the size of the layout (or, for that matter, the size of the individual interconnections). This is even the case when efficient and effective techniques for ordering the Gaussian elimination steps are used [10].

Thus, significant improvements of the efficiency of the FEM will be valuable. This paper will present such an improvement, which is based on partitioning the set of equations and solving each partition (which corresponds to a segment of an interconnection) independently. Since the assembly of the total result from the partial results can be done at zero computational cost, the total computational cost is reduced. Because for many layout styles the size of the sub-problems is bounded by a constant, we have obtained, for these layout styles, a linear time complexity.

Our algorithm makes use of so-called articulation nodes: an articulation node in a connected network is a node which, if deleted, would break the network into two or more pieces. Articulation nodes naturally exist in the network or can be introduced very easily without generating inaccuracies. They correspond to equipotential regions or lines in the interconnections. Effective partitions are then induced by these articulation nodes of the finite-element network. Our resistance extraction method then merely identifies and/or generates as many articulation nodes as possible, and subsequently preserves them during the Gaussian elimination process.

Apart from improved efficiency, another important advantage of the proposed method is that the resulting RC networks contain many fewer resistances and much better reflect the tree

33rd Design Automation Conference ®

DAC 96 - 06/96 Las Vegas, NV, USA

structure of the original VLSI interconnections. As a result, subsequent simulation or timing verification steps will be significantly more efficient and, in some cases, more accurate.

First, in Section 2 we give some background information on the finite-element method. Next, in Section 3, we describe how articulation nodes can be used to reduce the computational cost of the finite-element method. Then, in Section 4 and Section 5, we show how articulation nodes are actually introduced in the network. In Section 6, we discuss aspects of the implementation of the method in a layout-to-circuit extractor. Subsequently, in Section 7, we present results of the method that have been obtained using the layout-to-circuit extractor. Finally, in Section 8, we give some conclusions.

2 Background

2.1 The Finite-Element Method

For the purpose of resistance extraction, the finite-element method can be cast into a modeling method that directly produces an equivalent circuit model instead of a specific field solution. Thus, the result is an admittance matrix $\mathbf{Y}$ that relates the terminal currents to the terminal potentials. The terminals are formed by the pins and the connections of the wires to the active devices.

Without going into detail (but see, e.g. [1, 11]), we state that the finite-element method discretizes the interior of the interconnections and produces a system of equations that can be formulated as a matrix problem as in Equation (1):

$$\mathbf{A}\mathbf{x} = \mathbf{b}. \tag{1}$$

Here, $\mathbf{A}$ is a symmetric, positive definite and sparse matrix, $\mathbf{x}$ is a vector of unknown potentials and $\mathbf{b}$ is a vector of currents fixed by the boundary conditions. From the matrix $\mathbf{A}$ the matrix $\mathbf{Y}$ can be computed using Gaussian elimination [1, 11].

Instead of solving the system of equations Equation (1) and determining a resistance network only afterwards, from the resulting matrix $\mathbf{Y}$, we can begin with a circuit formulation of Equation (1). Such an interpretation of the FEM [1, 2, 11] initially produces a complex resistance network that models the resistive interconnections in detail. Subsequently, this network is reduced by a set of node eliminations. Each node elimination is in fact a Gaussian elimination step. If G_{ij}^k is the admittance between node i and j before the elimination of node k, and G_{ij}^{k+1} after the elimination of node k (we assume that node k is eliminated in step k), we have

$$G_{ij}^{k+1} = G_{ij}^k + \frac{G_{ik}^k G_{jk}^k}{\sum_{x \neq k} G_{kx}^k}. \tag{2}$$

Equation (2) states that the star network of the node that is eliminated is replaced by a full network (a *clique*) on the remaining nodes.

Layout-to-circuit extractors usually use this circuit or graph formulation [2]. It is easier to implement efficiently and is compatible with the data structures for the device connectivity. This paper will therefore use this circuit context, but the results will also be valid for an array-based implementation.

2.2 Delayed Frontal Solution Method

Equation (2) shows that a node k can be eliminated as soon as all the admittances G_{kx} connected to node k are known. Doing this for each node significantly reduces the amount of storage required when compared to first building the complete network and then doing the eliminations. Early elimination is in fact of paramount importance when large layouts must be handled.

The admittances connected to a node are all known when all the finite elements incident to that node have been processed (or assembled, in finite-element parlance). If the elements are processed systematically from one side of the layout to the other, the operations occur in a front that moves along the layout. Hence, this method is called the *Frontal Method*[12].

In a scanline based layout extractor, the frontal solution method can be efficiently combined with the minimum degree heuristic for optimizing the order of the Gaussian elimination operations using the so-called *Delayed Frontal Solution Method*[10]. Instead of being eliminated immediately when the node is ready, it is inserted into a *priority queue*. The queue has a fixed maximum capacity, and only when it is full a node with minimum degree is removed from the queue and eliminated from the network. Thus, at the cost of a moderate amount of extra memory, the time complexity is greatly reduced.

The delayed frontal solution method works conveniently together with the method presented in this paper, as will be shown in the results of Section 7.

3 Preservation of Articulation Nodes

Consider a connected resistance network. In such a network, an *articulation node* is a node which, if deleted, would break the network into two or more pieces. For each articulation node, this number of pieces is called its *articulation degree*. A network without articulation nodes is called *biconnected*, since there are (at least) two disjunct paths connecting each pair of nodes. If a network is not biconnected, it divides into *biconnected components*. A node which is not an articulation node is called a *regular node*. An articulation node can be part of more than one biconnected component, but a regular node is part of only one biconnected component.

For example, in Figure 1.a, node 4 is an articulation node. There are two biconnected components, namely (1, 2, 3, 4) and (4, 5, 6, 7) and the articulation degree of node 4 is 2 (its degree is 6).

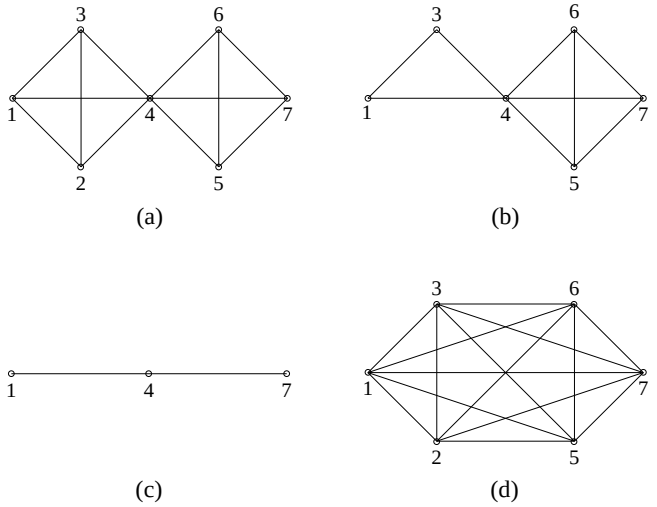

Figure 1: *(a) Resistance graph where node 4 is an articulation node. (b) The same graph when regular node 2 has been eliminated. (c) When regular nodes 3, 5 and 6 have also been eliminated. (d) The original graph in which the articulation node has been eliminated.*

The problem of Gaussian elimination in sparse interconnect networks is that of the generation of so-called *fill-ins*. The cost of eliminating a node with degree d is $O(d^2)$. When a node with a degree d is eliminated, it is replaced by a full network on the d remaining nodes, potentially generating an admittance between two nodes that were previously not directly connected. This may increase the degree, and thus the cost of subsequent elimination, of the remaining nodes. As a result, the time complexity of Gaussian elimination of N nodes is $O(N^p)$ with $p > 1$.

Our method is based on the following observation:

- *When a regular node is eliminated, it will not produce fill-ins in other biconnected components than its own.*

Hence, the biconnected components do not influence each other if only regular nodes are eliminated. Effectively, the elimination problem is then partitioned into a number of subproblems, one for each biconnected component. This partitioning is beneficial for the efficiency (see below) and we can easily enforce it by adopting the following strategy:

- *Only eliminate regular nodes.*

Then, after all regular nodes have been eliminated, the articulation nodes remain in the extracted network.

Because of the partitioning, the time complexity has been improved. Let n be the maximum size of a biconnected component. Then, the time complexity has been reduced to $O(\frac{N}{n} \times n^p)$ or $O(N n^{p-1})$. We see that it is advantageous to make n as low as possible or, equivalently, use as many articulation nodes as possible. Moreover, if n is bounded by a constant, the running time becomes $O(N)$: linear in the size of the layout.

The effect of the partitioning is that the number of fill-ins or, equivalently, the degree of a node when it is eliminated, is reduced. This is demonstrated in Figure 1. We assume that nodes 1 and 7 are terminal nodes that should not be eliminated. When starting with the resistance graph of Figure 1.a, Figure 1.b is obtained by eliminating regular node 2. No fill-ins are generated. When subsequently regular nodes 3, 5 and 6 are eliminated, the resistance graph of Figure 1.c is obtained. Now, node 4 can be eliminated at only a small cost. During this whole process, no nodes with a degree greater than 3 are eliminated. On the other hand, if, starting with the same resistance graph of Figure 1.a, articulation node 4 is deleted, the resistance graph shown in Figure 1.d will result. Node 4 has degree 6 when it is eliminated and the remaining nodes all have degree 5.

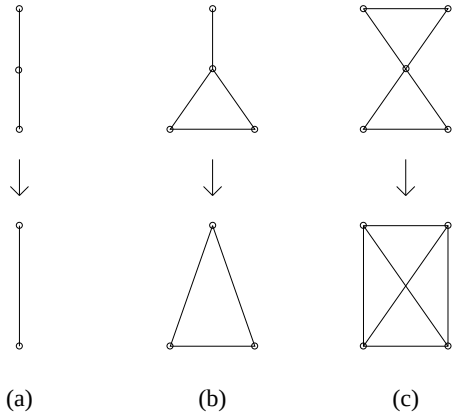

Figure 2: *Network simplification by elimination of articulation nodes that have a low degree.*

After all the regular nodes have been eliminated, it is possible to eliminate the articulation nodes. This will reduce the number of nodes but it can cause an explosion of the number of resistances. In particular, when all articulation nodes are eliminated, the result will be a full resistance graph on all the terminal nodes of the interconnect. When they are preserved, the result will be a much smaller (more nodes but many fewer resistances) network that better reflects the topology of the interconnect. Only for articulation nodes with a low degree, the number of resistances in the network will not explode when the node is eliminated. We therefore adopt the following strategy:

- *After all regular nodes have been eliminated, eliminate the articulation nodes that have a low degree.*

For example, in Figure 2.a, a series connection of 2 resistances is replaced by one resistor. In Figure 2.b, elimination of the articulation node also reduces the number of nodes and resistances. Finally, in Figure 2.c, elimination of the articulation node reduces the number of nodes but preserves the number of resistances.

In the next two sections, we describe how to introduce as many articulation nodes as possible, and in Section 7 we demonstrate that a linear time complexity is actually achieved in practice.

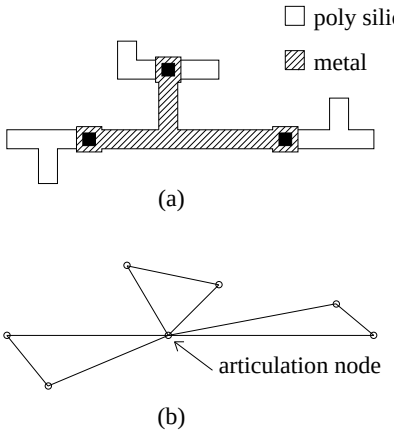

Figure 3: *(a) Piece of interconnect. (b) Corresponding resistance graph if metal resistances are not extracted and after elimination of all internal regular nodes: the metal wire has become an articulation node.*

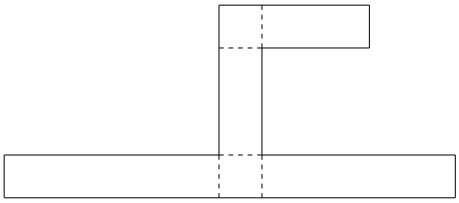

Figure 4: *Piece of interconnect subvided into rectangles.*

4 Equipotential Areas

Often, the resistances of metal wires can be neglected compared to the resistances of poly silicon and diffusion wires. In this case, the metal wires constitute equipotential areas that are represented in the resistance graph as articulation nodes. This is illustrated in Figure 3. The articulation nodes that are introduced by the equipotential areas have an articulation degree ≥ 2. In some cases, the articulation degree can be large (supply lines and clock lines).

5 Equipotential Lines

Another way to introduce articulation nodes in the resistance graph — that is especially important if metal wires are not modeled as equipotential areas — is by detecting equipotential lines for the interconnects. This is illustrated by the interconnect that is shown in Figure 4.

The interconnect in Figure 4 has been subdivided into rectangles. Some rectangles are connected at two opposite sides and are unconnected at the two other sides. One such a rectangle has been drawn in Figure 5. The distance between the sides that are connected is called the length L of the rectangle and the distance between the sides that are unconnected is called the width W of the rectangle.

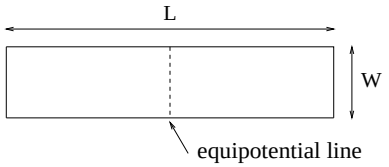

Figure 5: *Conductor rectangle that is connected to other conductor parts at the left side and at the right side.*

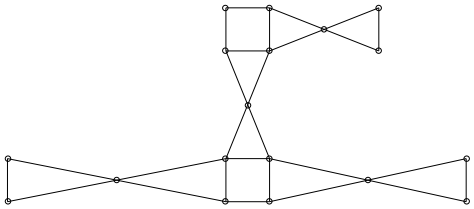

Figure 6: *Resistance graph for the interconnect in Figure 4 when equipotential lines are used to introduce articulation nodes.*

Clearly, for rectangles for which $L/W > k$, where $k \approx 1$ (see [4]), the current flow is approximately perpendicular to a line that is drawn parallel to the two sides that are connected and that splits the rectangle into two equal pieces. This is independent from the potential distribution along the connected edges. As a result, the line constitutes an equipotential line.

If the rectangle is split along its equipotential line, all nodes in the finite-element mesh that are on the equipotential line may be collapsed into one node. This introduces an articulation node with articulation degree 2. As an example, Figure 6 shows the resistance graph for the conductor in Figure 4.

In general, for practical layouts, when an equipotential line is generated for each rectangle for which $L/W > k$, a biconnected component represents a set of mesh nodes that is generated for a corner, a T-section or another irregular part of an interconnection. Therefore, typically the number of nodes in each biconnected component will be small and independent from size of the total layout.

6 Implementation in a Layout-to-Circuit Extractor

The above method for interconnect resistance extraction has been implemented in the layout-to-circuit extractor Space [13]. Besides resistances, Space also extracts the interconnect capacitances. Initially, the interconnect capacitances are assigned to the nodes in the finite-element mesh. Then, an Elmore time-constant preserving node reduction technique (which is an extension of the Gaussian elimination) is used to eliminate the nodes that are not desired in the final network. This produces a final RC network in which the distributed RC effects are accurately reflected[2, 9].

Table 1: *Extraction results for circuit "control" (1467 transistors) as a function of the number of equipotential areas, the number of equipotential lines and the queue size that are used.*

nr. of areas	nr. of lines	queue size	nr. of nodes	nr. of res.	time [min:sec]	memory [MByte]
0	0	0	3,446	33,283	30:04.7	10.40
0	0	500	3,443	32,294	2:38.7	6.74
0	8,123	0	3,480	3,829	1:55.7	3.33
0	8,123	500	3,470	3,816	59.3	3.16
1,282	0	0	2,702	2,771	47.6	2.19
1,282	0	500	2,702	2,770	37.6	2.06
1,282	8,123	0	3,017	2,988	35.0	2.33
1,282	8,123	500	3,009	2,977	35.0	2.62

7 Results

The results presented in this section were all obtained on a HP 9000/735 computer. The example circuits are digital CMOS circuits of different sizes. To compute the resistances, a coarse finite-element mesh was used similar as in Figure 4. The extraction times and memory usage given include all extraction operations, e.g. program start-up, element recognition and input-output operations. Besides interconnect resistances, contact resistances and MOS devices, also capacitances to ground were extracted. All extractions were done in flat extraction mode. Apart from the network simplifications that were obtained by introducing articulation nodes, in all cases "heuristics" were applied to simplify the extracted circuit by joining nodes connected by a small resistance and by removing large shunt resistances.

In Table 1, extraction results are presented for circuit "control". Resistances for this circuit were extracted without and with neglecting metal resistances (number of equipotential areas is 0 and number of equipotential areas > 0 respectively), without and with using equipotential lines, and using a normal frontal solution method (queue size is 0) or a delayed frontal solution method (queue size > 0). Note the effect that the introduction of articulation nodes, by neglecting metal resistances and by using equipotential lines, has on the extraction time and on the memory usage.

Besides extraction time and memory usage, also the number of nodes and the number of resistances in the extracted circuit are given in Table 1. One can see that the presence of articulation nodes significantly reduces the number of resistances in the output circuit.

Figure 7 shows the elimination degree frequency, the number of times that a node with a certain degree is eliminated, for circuit "control". From this figure we see that the frequency of occurrence of nodes with a high elimination degree decreases when articulation nodes are introduced. This explains the differences in extraction times in Table 1: since the elimination cost of each node k is $O(d_k^2)$, the total elimination cost becomes lower when equipotential lines and equipotential areas are used.

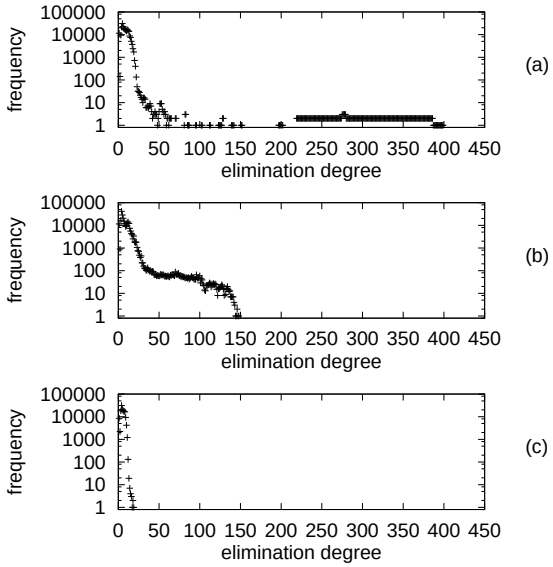

Figure 7: *Elimination degree frequency for circuit "control" when not using articulation nodes (a), when using equipotential lines to introduce articulation nodes (b) and when using equipotential lines and equipotential areas to introduce articulation nodes (c).*

In Figure 8 extraction results are presented for CMOS circuits of different sizes. In all cases the delayed frontal solution method was used. From the results we see that when no articulation nodes are used, the number of resistances, extraction time and memory usage increase rapidly as a function of the size of the circuit. In fact, when no articulation nodes are used, it is not possible to extract circuits containing more than 6000 transistors because of memory swapping. When articulation nodes are used, the extraction time is linearly dependent on the size of the circuit and the relation between memory usage and circuit size is better than linear.

8 Conclusions

We have shown that the introduction of articulation nodes in the finite element network and their preservation during the Gaussian elimination phase, greatly improves the efficiency of the finite element method for resistance extraction of VLSI interconnections.

In particular, we have demonstrated the following results:
- Finite element based resistance extraction of VLSI interconnections (including metal resistance extraction) can be done in time linear in the size of the layout (number of transistors).
- The amount of memory that is needed, is better than linear in the size of the layout.
- The topology of the extracted resistance network more

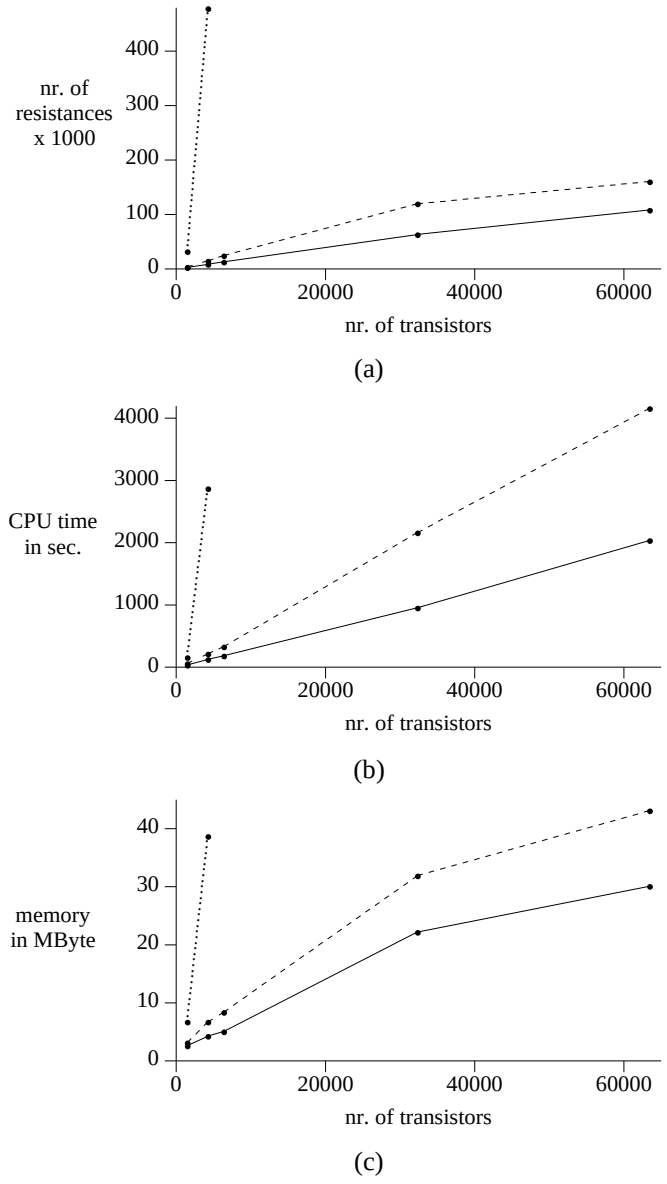

Figure 8: *Number of extracted resistances (a), extraction time (b) and memory usage (c) as a function of circuit size. Dotted line = no articulation nodes, dashed line = using equipotential lines, solid line = using equipotential lines and equipotential areas.*

closely matches the tree structure of the interconnections. This will improve the accuracy of subsequent switch-level simulation or timing verification. (These programs often cannot handle parallel signal paths correctly.)

- The number of resistances in the final extracted network is greatly reduced.
- As implemented in the Space [13] layout to circuit extractor, the method enables accurate extraction of RC interconnection models of large VLSI layouts using comparatively little CPU time and memory. For example, a complete, flat, extraction of a 63,416-transistor circuit including all polysilicon, diffusion and contact resistances took only 34 minutes and used 30 Mbytes.

Acknowledgements

This research is sponsored in part by the Dutch Technology Foundation (STW) under grant nr. DEL 11.2450 and grant nr. DEL 22.2810, and has been carried out in the context of DIMES, the Delft Institute of Microelectronics and Submicron Technology.

References

[1] T. Mitsuhashi and K. Yoshida, "A Resistance Calculation Algorithm and its Application to Circuit Extraction," *IEEE Trans. CAD*, vol. CAD-6, no. 3, pp. 337–345, 1987.

[2] A. J. van Genderen and N. P. van der Meijs, "Extracting Simple But Accurate RC Models for VLSI Interconnect," in *Proc. ISCAS-88*, (Helsinki, Finland), pp. 2351–2354, June 1988.

[3] D. Stark, "Analysis of Power Supply Networks in VLSI Circuits," Tech. Rep. WRL-TR-91-3, Digital Equipment Corporation, Western Research Laboratory, 1991.

[4] S.P. McCormick, "EXCL: A circuit extractor for IC designs," in *Proc. 21st Design Automation Conference*, pp. 616–623, 1984.

[5] I.S. Duff, A.M. Erisman, and J.K. Reid, *Direct Methods for Sparse Matrices*. Oxford Science Publications, 1986.

[6] M. Horowitz and R. W. Dutton, "Resistance extraction from mask layout data," *IEEE Trans. on CAD*, vol. CAD-2, pp. 145–150, July 1983.

[7] P.M. Hall, "Resistance Calculation for Thin Film Patterns," *Thin Solid Films*, vol. 1, pp. 277–295, 1967/1968.

[8] B.R. Chawla and H.K. Gummel, "A Boundary Technique for Calculation of Distributed Resistance," *IEEE Trans. Electron Devices*, vol. ED-17, pp. 915–25, 1970.

[9] M. G. Harbour and J. M. Drake, "Calculation of Signal Delay in Integrated Interconnections," *IEEE Trans. on Circuits and Systems*, vol. CAS-36, pp. 272–276, Feb. 1989.

[10] N. P. van der Meijs and A. J. van Genderen, "Delayed Frontal Solution for Finite-Element based Resistance Extraction," in *Proceedings of the 32nd Design Automation Conference*, (San Francisco, USA), pp. 273–278, June 1995.

[11] A. J. van Genderen, *Reduced Models for the Behavior of VLSI Circuits*. PhD thesis, Delft University of Technology, Delft, the Netherlands, Oct. 1991.

[12] B.M. Irons, "A Frontal Solution Scheme for Finite Element Analysis," *Numer. Meth. Engng*, vol. 2, pp. 5–32, 1970.

[13] N. P. van der Meijs and A. J. van Genderen, "Space User's Manual," Tech. Rep. ET-NT 92.21, Delft University of Technology, Dept of EE, Delft, NL, Apr. 1992.

[14] W.S. Scott and J.K. Ousterhout, "Magic's Circuit Extractor," in *Proc. 22nd Design Automation Conference*, pp. 286–292, 1985.