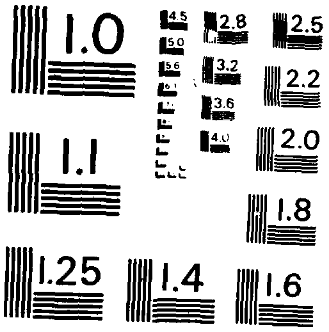
AD-A142 381 MODELING AND EXTRACTION OF INTERCONNECT PARAMETERS IN VERY-LARGE-SCALE INTEGRATED CIRCUITS(U) ILLINOIS UNIV AT URBANA COORDINATED SCIENCE LAB C P YUAN AUG 83 1/3

UNCLASSIFIED R-994 N00014-79-C-0424 F/G 9/5 NL



MICROCOPY RESOLUTION TEST CHART

NATIONAL BUREAU OF STANDARDS-1963-A

12

REPORT R-994 AUGUST 1983 UILU-ENG 83-2215

COORDINATED SCIENCE LABORATORY

MODELING AND EXTRACTION OF INTERCONNECT PARAMETERS IN VERY-LARGE-SCALE INTEGRATED CIRCUITS

CHEN-PING YUAN

APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED.

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER R-994	2. GOVT ACCESSION NO. AD-A242 382	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Modeling and Extraction of Interconnect Parameters in Very-Large-Scale Integrated Circuits		5. TYPE OF REPORT & PERIOD COVERED Technical Report
		6. PERFORMING ORG. REPORT NUMBER R-994; UILU-ENG 83-2215
7. AUTHOR(s) Chen-Ping Yuan		8. CONTRACT OR GRANT NUMBER(s) N00014-79-C-0424
9. PERFORMING ORGANIZATION NAME AND ADDRESS Coordinated Science Laboratory University of Illinois at Urbana-Champaign Urbana, IL 61801		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Joint Services		12. REPORT DATE August 1983
		13. NUMBER OF PAGES 193
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

very large scale integrated circuits
extraction of circuit from layout
electrical models for interconnect circuit

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The increased complexity of the very large scale integrated circuits (VLSI) has greatly impacted the field of computer-aided design (CAD). One of the problems brought about is the interconnection problem.

In this research, the goal is two fold. First of all, a more accurate numerical method to evaluate the interconnect capacitance, including the coupling capacitance between interconnects and the fringing field capacitance, was investigated, and the integral method was employed. Two FORTRAN programs "CAP2D" and "CAP3D" based on this method were developed. Second, a PASCAL extraction program emphasizing the extraction of interconnect parameters was developed. It employs the "cylindrical approximation formula" for the self-capacitance of a single interconnect and other simple formulas for the coupling capacitances derived by a "least square method". The extractor assumes only Manhattan geometry and NMOS technology. Four-dimensional binary search trees are used as the basic data structure.

MODELING AND EXTRACTION
OF INTERCONNECT PARAMETERS
IN VERY-LARGE-SCALE INTEGRATED CIRCUITS

BY

CHEN-PING YUAN

B.S., National Taiwan University, 1975
M.S., University of Illinois, 1979

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1983

Urbana, Illinois

MODELING AND EXTRACTION
OF INTERCONNECT PARAMETERS
IN VERY-LARGE-SCALE INTEGRATED CIRCUITS

Chen-Ping Yuan, Ph.D.
Department of Electrical Engineering
University of Illinois at Urbana-Champaign, 1983

The increased complexity of the very large scale integrated circuits (VLSI) has greatly impacted the field of computer-aided design (CAD). One of the problems brought about is the interconnection problem.

In this research, the goal is twofold. First of all, a more accurate numerical method to evaluate the interconnect capacitance, including the coupling capacitance between interconnects and the fringing field capacitance, was investigated, and the integral method was employed. Two FORTRAN programs "CAP2D" and "CAP3D" based on this method were developed. Second, a PASCAL extraction program emphasizing the extraction of interconnect parameters was developed. It employs the "cylindrical approximation formula" for the self-capacitance of a single interconnect and other simple formulas for the coupling capacitances derived by a "least square method". The extractor assumes only Manhattan geometry and NMOS technology. Four-dimensional binary search trees are used as the basic data structure.

ACKNOWLEDGEMENTS

To Professor T. N. Trick, my advisor, I owe my deepest thanks. Without his guidance, support and encouragement, it would not have been possible to complete my study. I am indebted to Professor I. N. Hajj, Professor J. D. Dyson and Professor D. J. Brown, who agreed to be on my committee and made many valuable suggestions. Also I would like to express my gratitude to the Joint Services Electronics Program (U. S. Army, U. S. Navy and U. S. Air Force) and the IBM Corporation for their financial support during my graduate studies. Finally, to my family, it is their love, understanding and support which made the lonely and sometimes anguished journey more bearable.

TABLE OF CONTENTS

CHAPTER Page

1. INTRODUCTION 1

2. NUMERICAL COMPUTATION OF INTERCONNECT CAPACITANCE 6

3. APPROXIMATION FORMULAS FOR INTERCONNECT CAPACITANCES 63

4. NETWORK EXTRACTOR 91

5. CONCLUSION 137

APPENDICES

A. THE VALIDITY OF THE LUMPED CIRCUIT MODEL 140

B. CAP USER'S MANUAL 145

C. COMPUTATION OF TWO-DIMENSIONAL KERNEL MATRIX ELEMENTS 165

D. EVALUATION OF THREE-DIMENSIONAL KERNEL MATRIX ELEMENTS 176

E. SPICE INPUT DECK FOR FULL ADDER I 180

F. SPICE INPUT DECK FOR FULL ADDER II 183

REFERENCES 186

VITA 193

CHAPTER 1

INTRODUCTION

The advent of very-large-scale integrated (VLSI) circuits as a result of rapid progress has greatly impacted on the various aspects in microelectronics. For example, as feature dimensions approach the submicron range, the "short-channel effects" dominate the physics of the devices. This has brought forth more research and studies into device physics, processing technologies, material science, etc. On the other hand, the exponential growth of IC complexity may push to the extent that the future progress in circuit integration will no longer be limited only by technology, but also, and to a much greater extent, by our ability to reduce the astronomical design time. This concern has spawned research and studies in computer-aided-design (CAD), computer architecture, design methodology, design automation, etc. [1-49]. All these efforts are aimed at further advancing the progress in microelectronics that we have enjoyed for the past two decades. Hopefully, through these studies and with the computational power VLSI circuits have brought us, we can "bootstrap" ourselves to the next generation of technologies.

A comprehensive CAD system consisting of various tools has become indispensable in VLSI circuit design. A part of the CAD system is design verification which shows that a given mask geometry will perform the desired digital functions as originally designed.

Extraction, the first step in design verification, transforms the geometric information on the masks into circuit information. Afterwards a comparison can be made with the original design, or a circuit simulation can be performed to check the responses of the circuit.

One of the problems that the increased complexity has brought about in CAD is the interconnection problem. Not only does the routing among a myriad of devices on the chip become complicated, difficult and time-consuming, but also the electrical parameters of these very interconnects have gradually grown into an integral part of the circuits. The increased complexity has augmented the lengths of interconnects on the chip. Moreover, the reduced dimensions of the interconnects themselves increase the resistances of the lines, and the fringing capacitances are no longer negligible. Consequently, it is necessary to include these interconnect parameters in the circuit model. It is on this aspect that this thesis concentrates.

First of all, an equivalent circuit model for the interconnects on the chip has to be established. A T-type lumped circuit model is used in this thesis for the interconnects assuming silicon technology. In Appendix A, criteria for the validity of the lumped circuit model are derived from transmission line theory. It is shown that the lumped circuit models are generally applicable under the present technologies.

In the first part of the thesis, a more accurate account of the interconnect capacitance by numerical methods is considered. Both the differential method, e.g., the finite difference method, and the integral method (by way of Green's functions) have been studied. Based on the applications of both methods to solve various problems in the literature [2-10], combined with the experience in this study, it was found that the integral method seems to offer more advantages in the flexibility of programming and in computation time. Therefore, it was employed for the numerical computation of two- and three-dimensional interconnect capacitances. Basically the method follows the "method of moments" proposed by Harrington [11] and is similar to those in [8-10]. Both testing functions and basis functions are chosen the same as pulse-type functions, which are constants over the subdivisions, and variable-length subsections are divided according to the roots of Chebyshev polynomials. It was found that this approach offered more accuracy than a comparable number of constant subdivisions or using delta-type testing functions, and the positive-definiteness of the resulting kernel matrix also guaranteed a solution. Some results will be presented in Chapter 2.

Next, with a view to easily incorporating these capacitances in the extraction program for design verification, simple approximation formulas were sought. An approximation formula for the self-capacitance of a single interconnect was first derived. It is based on the known expression of the capacitance of a cylinder over a ground plane. Employing this expression to approximate the fringing part of the self-capacitance, we have a reasonable approximation formula with an accuracy within 10% of the numerically computed values, and it is named "cylindrical approximation formula". As to the coupling capacitances between interconnects, a "least square fit" method on the data obtained from the numerical computation was employed to find approximation formulas. A few formulas have been obtained for different interconnect configurations, e.g., coupling capacitances between two parallel interconnects or two pe ɹndicular cross-over lines.

The second part of the thesis is mainly ɹncerned with the extraction program. It was intended, if possible, to be part of a hierarchical CAD system which supports a number of functions at each level in the design hierarchy with different tools. Hence, the output form of the extractor is in a SPICE input compatible format such that the circuit simulation can be performed without the data conversion. The basic data structure in the extractor is 4-d binary search trees of geometrical rectangles on the masks. The output includes not only the pertinent information of transistors, e.g., channel width and channel length, but also the parameters of interconnects. The resistances and the capacitances, both self- and coupling capacitances, are reported.

In summary, Chapter 2 focuses on the numerical computation of the interconnect capacitance, both in two and three dimensions. It comprises the detailed formulation of the Green's functions, the even-odd analysis to find the coupling capacitances in multiconductor cases, the integral method and the comparison with the differential method, and a few examples. A user's manual for the programs "CAP2D", "CAP3D" which execute the aforementioned numerical computation is described in Appendix B. Chapter 3 covers the approximation formulas of capacitances. The "cylindrical approximation formula" is derived, and other approximation formulas for the coupling capacitances are also discussed. In Chapter 4, the network extraction program's data structure and the procedure of extracting transistors and interconnect parameters are detailed; its characteristics, limitations and some examples will also be discussed. Finally, the conclusions are in Chapter 5.

CHAPTER 2

NUMERICAL COMPUTATION OF INTERCONNECT CAPACITANCE

2.1. Introduction

The present very large scale integration circuits lead to very large and complex systems with small physical dimensions. Not only do the devices have small dimensions which require special attention, but the interconnects also become important and have to be reckoned with in the circuit simulation. A better understanding of the electrical characteristics of the interconnect, especially the capacitance, is essential in the design and analysis of today's integrated circuits.

As a general rule of thumb, the highest frequency components contained in the signals in most present large scale integrated circuits correspond to wavelengths well exceeding the physical dimensions of the circuit elements. Hence, lumped circuit models can be used for the electrical analysis of the circuits. G. Bilardi et al. [12] evaluated various circuit models for interconects in VLSI circuits and it was concluded that both current and the projected silicon technologies in the near future fall within the realm of the capacitance model. That is to say that a dispersive line can be replaced by a capacitance proportional to its length. From another point of view, the lumped circuit model can be considered as a first-order approximation of the transmission line model as discussed in Appendix A.

Another important aspect in the electrical analysis of the integrated circuit is the "general impedance level". It is defined in [13] as the lossless characteristic impedance of the average connection in the system. Since most digital integrated circuits are based on MOSFET devices which are typically of higher impedance than the general impedance level, voltage and charge become the "state" variables in the analysis. Consequently, capacitance emerges as the dominant circuit element. The deciding factor of the performance of most digital circuits is the time delay through the system, and interconnect capacitance certainly plays an important role.

A cross-sectional view of a typical integrated circuit is shown in Fig. 2.1. Since the channel-stopping ion implantation is usually situated beneath the field oxide where the interconnect is generally placed, it can be considered as the ground plane as shown in the computation model in Fig. 2.2. In general cases, there can be three different dielectric media, and interconnects can reside inside the oxide layer or the passivation layer as shown in Fig. 2.2. The topmost layer is assumed to be always the free space. In order to reduce the computational effort, we can also consider the interconnect capacitance in a two layer or a homogeneous medium if the layers in question are thick enough. The following discussion of the numerical computational method of the interconnect capacitance is based on this model as shown in Fig. 2.2.

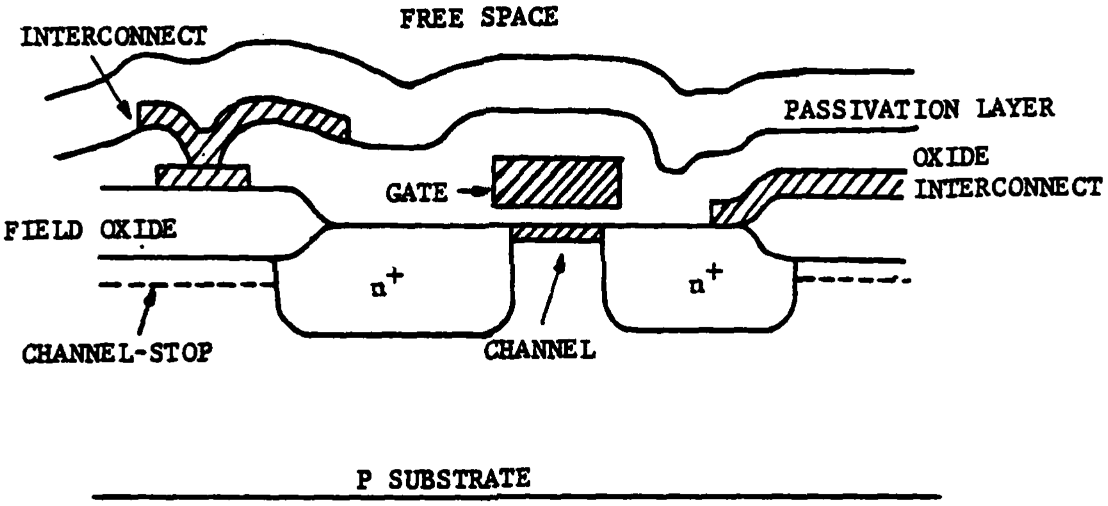


Figure 2.1 A cross-sectional view of a typical integrated circuit.

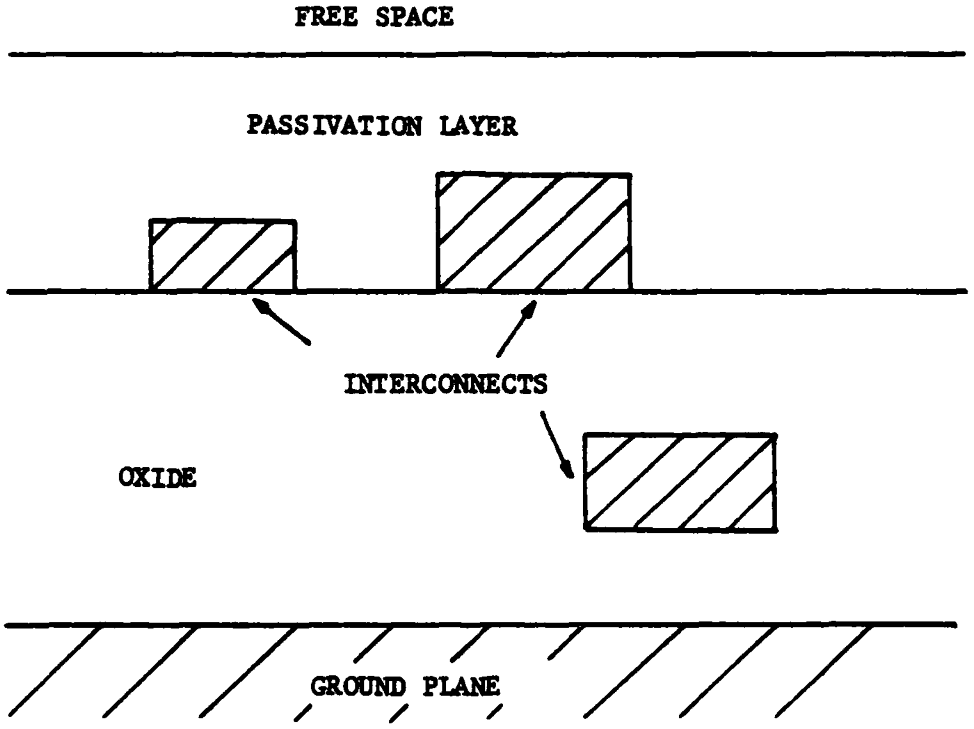


Figure 2.2 Computation model of interconnect capacitance.

2.2. Numerical Computation Methods of Interconnect Capacitances

The computation of capacitances has appeared in the past for various applications, e.g., the calculation of the characteristic impedance of microstrip transmission lines is one prime example [42-45]. Though the emphasis and the assumptions are different in different applications, the methods are still applicable in the computation of interconnect capacitance. In summary, the available computation methods of interconnect capacitance can be categorized into three types, namely: analytic, differential and integral methods.

The analytic method, in effect, can calculate the capacitance in closed form and no numerical analysis or discretization is required. This will be the subject of the next chapter so its discussion will be deferred. In this chapter we will concentrate on the numerical methods, i.e., the differential and integral methods.

Let us first consider the differential method. The finite difference method for solving boundary value problems is the prime example of a differential method. It starts with Poisson's differential equation, defines the boundary of the problem in question, and then sets up the boundary conditions. After solving all the potentials on the grid points within the boundaries, we can calculate the charges on the surface of the interconnect and then the capacitances accordingly. For example, the method employed in [2,3] can be categorized as the differential method. It is generally easy and straightforward to formulate. Multiple layers of media or irregular boundaries can be accommodated without too much increase in computation. However, if the region in consideration is an open boundary as shown in Fig. 2.2, a suitable boundary condition has to be defined. In [2] a "reflective"- or Neumann-type boundary condition at a reasonable distance from the source was defined to simulate the open boundary. In [3] an ad hoc arrangement on the boundary grids was set up in the numerical process to take care of the open boundary. Though there are other ways to reasonably simulate the open boundary, they may compromise the accuracy of the computation or increase the computation time. This is one disadvantage of the differential method. Other disadvantages may be attributed to the rather large system of linear algebraic equations of grid potentials and the difficulty of extending it to three-dimensional cases. Depending on the grid and the regularity of the boundaries and the source regions, this matrix may not lend itself to some special techniques in numerical linear algebra. Iterative methods like Gauss-Seidel may be too time-consuming and slowly convergent. In three-dimensional cases, the number of discretized grids may be too large to handle in the finite difference method.

On the other hand, the integral method starts with the Green's function of the Poisson's equation, and the integral equation of the convolution of the Green's function with the charge density function is solved. Then the capacitances are calculated after the charge densities on the interconnect have been found. In [4] the Green's function for two-dimensional interconnects in two layers of media was derived, and it will be shown in a latter section. In [6] the three-dimensional Green's function was formulated for the computation of irregular conductors over dielectrics.

In comparing these two methods, the differential method can be thought of as solving the problem "microscopically", and the integral method as "macroscopically". Because the system of linear algebraic equations generated in the finite difference scheme relates only local neighboring points in the grid, the resulting matrix tends to be large but sparse. The effects of the boundaries and the sources on the grid potentials are reflected only after the system of equations is solved by, for example, the Gauss-Seidel method. While, on the other hand, the effects of the boundaries are already "integrated" in the Green's function in the integral method; thus, the region in question only concerns the source region, and the resulting kernel matrix is small but full in this case. There are other differences between these two methods, and they are summarized in Table 2.1.

One of the important characteristics of these two methods is that the capacitance computed by the integral method will be a lower bound to the true value and that by the differential method will be an upper bound. This is parallel to the computation of the characteristic impedance of a waveguide discussed in [41], where the variational properties of the two approaches were also discussed. As in [41], it would be ideal to compute the capacitance twice, once by each method and take the average. However, there may be too much computation involved this way. Besides, the accuracy of the results

TABLE 2.1 Comparison of the Differential and Integral Methods.

Item of Comparison	Differential Method	Integral Method
Basic equation	Poisson's differential equation	Green's function and its convolution integral
"State" variable*	Potentials	Charge densities
Discretized region	Large region enclosing the source and the media around it	Only the surface of the source region
Multiple layers of media	Easy to accommodate without too much increase in computation	Infinite summation in Green's function causes a substantial increase in computation
Multiple conductors	Cannot take advantage of the even-odd mode excitation, substantial increase in computation; hard to automate in programming	Easy to take advantage of the even-odd mode excitation, and easy to automate the input of the dimensions and positions of the conductors
Automatic mesh generation	Harder to implement in a program	Easy to implement
Three-dimensional case	Much more complicated than two-dimensional counterpart	Not too much more complicated
Solution method	Solution of a large linear algebraic equation, i.e., Gauss-Seidel	Moment method
Boundaries shape	Can take care of more irregular boundaries provided the grid is refined enough	Rigid, regular boundaries, at most slant side walls and plane interfaces
Open boundaries	Some approximation or other mechanism has to be devised	No problem with open boundaries
Characteristics of the result**	Upper bound on the true value	Lower bound on the true value

* State variable means the unknown in the basic equation first was solved by either the differential or integral method.

** See Ref. [41] by Collin for a complete discussion.

can be increased by refining the grid in either method. In comparing the pros and cons between these two methods, the integral method seems to be the better choice, especially in the case of homogeneous medium and multiple conductors. It is easy to program for multiple conductors with diverse geometries and positions; the discretized region only concerns the conductor surfaces, which reduces the number of linear algebraic equations. More important, we can take advantage of the LU factorization of the kernel matrix in solving charge densities for the multiple conductor case. However, two major disadvantages of the integral method are in the cases of multiple layers of media and irregular shapes of the interconnects and the boundaries. If accurate numerical computation is imperative for those cases, the finite element method, one of the differential methods, may be the only viable solution.

2.3. Formulation of Multi-Conductor Capacitance Computation

In this section, a general formulation for computing the self- and the coupling capacitances among multiple conductors is described. Either the differential or the integral method has to be associated in this frame of formulation to compute the capacitances. First the general case of N conductors is derived, then two examples of N = 3 and N = 4 are shown. Special consideration is required for the case N = 4.

2.3.1. General formulation for N conductors

Consider a system of N conductors over a ground plane as shown in Fig. 2.3 where N = 3 is depicted. All the capacitances, both self- and coupling, are shown. The algebraic relationship between the charges Q_i, i=1,2,...,N and the potentials ϕ_i, i=1,2,...,N on the conductors can be expressed as [14]

$$\begin{array}{lcl} Q_1 & = & C_{11}\phi_1 + C_{12}(\phi_1-\phi_2) + \cdots\cdots + C_{1n}(\phi_1-\phi_n) \\ Q_2 & = & C_{21}(\phi_2-\phi_1) + C_{22}\phi_2 + \cdots\cdots + C_{2n}(\phi_2-\phi_n) \\ \vdots & & \qquad\qquad\vdots \qquad\qquad\qquad\qquad \vdots \\ Q_n & = & C_{n1}(\phi_n-\phi_1) + C_{n2}(\phi_n-\phi_2) + \cdots + C_{nn}\phi_n \end{array} \qquad (2.1)$$

Note that the potentials ϕ's are referenced with respect to the ground plane, and the capacitances are symmetric, i.e., $C_{ij} = C_{ji}$, for $1 \leq i,j \leq N$, because of reciprocity.

An analysis named "even-odd mode analysis" is employed to solve the capacitances C_{ij}'s in Eq. (2.1). Different potential patterns are first assigned on the conductors, even mode or odd modes, then the corresponding charges can be expressed algebraically in terms of the capacitances as will be shown in the N = 3 and N = 4 examples. Solving for the charges numerically in the respective modes by either the differential or the integral method, then we can find easily all the capacitances.

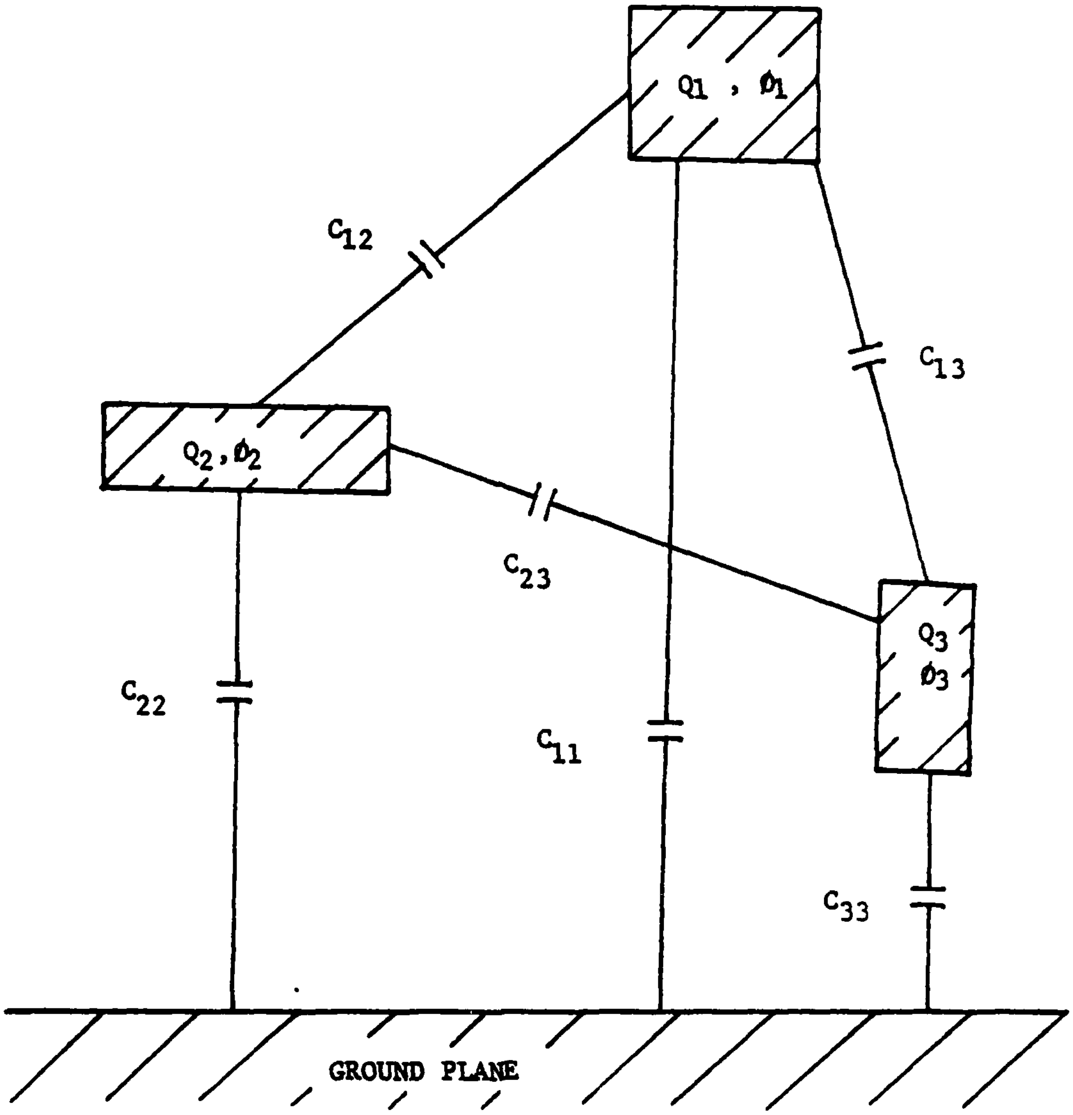


Figure 2.3 Multi-Conductor system. (N = 3 in this case).

(i) Even mode : In the even mode the potentials are assigned as $\phi_1 = \phi_2 = \dots = \phi_n = 1$. From Eq. (2.1) the self-capacitances can be written as $C_{ii} = Q_i^{(e)}$, i=1,2,...,N. The superscript on the charges indicates that those charges are evaluated under the even-mode potential pattern. All the self-capacitances are determined by this even-mode excitation.

Since there are N(N-1)/2 coupling capacitances left to be determined, it is necessary to generate the same number of independent algebraic equations from the odd-mode potential patterns. It is a matter of choice to assign these odd-mode potential patterns. A systematic procedure is used here. Except for the case N = 4, the procedure employed here produces a system of N(N-1)/2 linear independent algebraic equations which can be programmed easily and then solved uniquely for the coupling capacitances.

(ii) Odd mode 1 : the potentials are assigned as $\phi_1 = \phi_2 = 1$, and all the other potentials equal -1. Solving Eq. (2.1), we have

$$Q_1^{(1)} = C_{11} + 2(C_{13}+C_{14}+\dots+C_{1n})$$

$$Q_2^{(1)} = C_{22} + 2(C_{23}+C_{24}+\dots+C_{2n})$$

The sum of the above two equations is chosen to be one of the N(N-1)/2 independent equations. Since C_{11} and C_{22} have been found in the even-mode analysis, the sum of the two equations can be expressed as

$$C_{13} + C_{14} + \ldots + C_{1n} + C_{23} + C_{24} + \ldots + C_{2n}$$

$$= 0.5(Q_1^{(1)} + Q_2^{(1)} - Q_1^{(e)} - Q_2^{(e)}) \qquad (2.2)$$

The superscript 1 on the charges here indicates the first odd-mode excitation.

Other odd-mode patterns can be obtained as in (ii) by assigning two of the N potentials 1 volt, and all the other potentials -1 volt. Since there are totally C(N,2) = N(N-1)/2 combinations of this type, a total of N(N-1)/2 linear algebraic equations as Eq. (2.2) can be generated. It can be shown that the determinants of these systems of algebraic equations are nonzero for N = 3 to 10 except N = 4. So the linear independence is achieved and a unique solution is guaranteed.

2.3.2. Example for N = 3

First Eq. (2.1) is rewritten in the following form for N = 3.

$$Q_1 = C_{11}\phi_1 + C_{12}(\phi_1-\phi_2) + C_{13}(\phi_1-\phi_3) \qquad (2.3a)$$

$$Q_2 = C_{12}(\phi_2-\phi_1) + C_{22}\phi_2 + C_{23}(\phi_2-\phi_3) \qquad (2.3b)$$

$$Q_3 = C_{13}(\phi_3-\phi_1) + C_{23}(\phi_3-\phi_2) + C_{33}\phi_3 \qquad (2.3c)$$

The even- and odd-mode excitations are listed as follows :

(i) Even mode : let $\phi_1 = \phi_2 = \phi_3 = 1$ then we have

$$C_{11} = Q_1^{(e)} \quad ; \; C_{22} = Q_2^{(e)} \quad ; \quad C_{33} = Q_3^{(e)}$$

(ii) Odd mode 1 : let $\phi_1 = \phi_2 = 1$, and $\phi_3 = -1$, then we have

$$C_{13} + C_{23} = 0.5(Q_1^{(1)} + Q_2^{(1)} - Q_1^{(e)} - Q_2^{(e)})$$

(iii) Odd mode 2 : let $\phi_1 = \phi_3 = 1$, and $\phi_2 = -1$, then we have

$$C_{12} + C_{23} = 0.5(Q_1^{(2)} + Q_3^{(2)} - Q_1^{(e)} - Q_3^{(e)})$$

(iv) Odd mode 3 : let $\phi_2 = \phi_3 = 1$, and $\phi_1 = -1$, then we have

$$C_{12} + C_{13} = 0.5(Q_2^{(3)} + Q_3^{(3)} - Q_2^{(e)} - Q_3^{(e)})$$

It can be easily shown that the determinant of the system of the above three linear algebraic equations from (ii),(iii) and (iv) is 2. Hence, the coupling capacitances C_{12}, C_{13} and C_{23} are solved in terms of the corresponding charges as

$$C_{12} = 0.25(Q_1^{(2)}-Q_1^{(1)}+Q_2^{(3)}-Q_2^{(1)}+Q_3^{(2)}+Q_3^{(3)}-2Q_3^{(e)}) \qquad (2.4a)$$

$$C_{13} = 0.25(Q_1^{(1)}-Q_1^{(2)}+Q_3^{(3)}-Q_3^{(2)}+Q_2^{(1)}+Q_2^{(3)}-2Q_2^{(e)}) \qquad (2.4b)$$

$$C_{23} = 0.25(Q_2^{(1)}-Q_2^{(3)}+Q_3^{(2)}-Q_3^{(3)}+Q_1^{(1)}+Q_1^{(2)}-2Q_1^{(e)}) \qquad (2.4c)$$

Note that the subscript i on the charge $Q_i^{(j)}$ indicates the number of the conductor the charge is on, while the superscript j shows the number of the odd-mode excitation this charge is extracted from. Since there are other ways to define the odd-mode potential patterns, the coupling capacitance formula in Eq. (2.4) is not unique. For example, the capacitance formula in [5] is the result of another way to specify the odd-mode excitation and not the same as that in Eq. (2.4).

2.3.3. Special consideration for N = 4

The procedure described in Sec. 2.3.1 would result in a system of linearly dependent equations for the case N = 4. Another way to specify the even-odd mode excitation has to be devised here. The even-mode excitation can still be the same, while some other odd-mode excitations have to be defined. Instead of summing the corresponding equations resulting from the odd-mode excitations as mentioned above, they can be treated as separate equations. The resulting system of linear algebraic equations can be shown to be independent. So the even-odd mode analysis for the N = 4 case can be summarized as follows :

even mode : $\emptyset_1 = \emptyset_2 = \emptyset_3 = \emptyset_4 = 1$

$$C_{11} = Q_1^{(e)} \qquad C_{22} = Q_2^{(e)}$$

$$C_{33} = Q_3^{(e)} \qquad C_{44} = Q_4^{(e)}$$

odd mode 1 : $\phi_1 = \phi_2 = 1,\ \phi_3 = \phi_4 = -1$

$$Q_1^{(1)} = C_{11} + 2(C_{13}+C_{14}) \tag{2.5}$$

$$Q_2^{(1)} = C_{22} + 2(C_{23}+C_{24}) \tag{2.6}$$

odd mode 2 : $\phi_1 = \phi_3 = 1,\ \phi_2 = \phi_4 = -1$

$$Q_1^{(2)} = C_{11} + 2(C_{12}+C_{14}) \tag{2.7}$$

$$Q_3^{(2)} = C_{33} + 2(C_{23}+C_{34}) \tag{2.8}$$

odd mode 3 : $\phi_1 = \phi_4 = 1,\ \phi_2 = \phi_3 = -1$

$$Q_1^{(3)} = C_{11} + 2(C_{12}+C_{13}) \tag{2.9}$$

$$Q_4^{(3)} = C_{44} + 2(C_{24}+C_{34}) \tag{2.10}$$

Equations (2.5) to (2.10) constitute a system of six algebraic equations with determinant 4. Thus, the six coupling capacitances are uniquely determined. Note that if the general procedure in Sec. 2.3.1 is employed, then the next odd-mode pattern will be $\phi_2 = \phi_3 = 1$, $\phi_1 = \phi_4 = -1$. This is the exact complement to odd-mode 3 listed above. Consequently, the summing of the corresponding equations will result in an identical left-hand side from both excitations and a dependent system of equations. For cases other than N = 4, it is not possible to find pairs of "complementary excitations"; hence, the equations are independent.

2.4. Integral Method

In this section, a detailed discussion of the integral method is given. The method of moments [11] is formulated, along with the incorporation of the "even-odd mode analysis" discussed in Sec. 2.3. The two- and three- dimensional Green's functions in multiple layers of media will be derived in the next two sections, and the detailed expressions used to compute the kernel matrix will be given in Appendices C and D.

2.4.1. Formulation of moment method

Assume that the Green's function to the Poisson's equation is expressed as $G(\bar{r},\bar{r}')$. In the derivation of this Green's function, the boundary conditions have already been taken into account. In analogy to system analysis, the Green's function can be considered as the impulse response, and the potential in Poisson's equation can then be expressed as a convolution integral :

$$\phi(\bar{r}) = \int_S \sigma(\bar{r}')G(\bar{r},\bar{r}')\ d\bar{r}' \qquad (2.11)$$

In Eq. (2.11) ϕ is the potential and σ is the charge density on the source surface whose area is designated as S. Note that the dummy variable in the integral $\bar{r}'$ indicates the coordinate of the source region, and $\bar{r}$ is the observation point. Since potentials on the source region are decided by the even-odd mode excitation as discussed in the previous section, and the Green's function is also known, the only unknown variable in Eq. (2.11) is the charge density function $\sigma(\bar{r}')$. Therefore, it is an integral equation, and the method of moments can be employed to solve it.

First, assume that the unkonwn charge density function $\sigma(\bar{r}')$ can be expressed in terms of a set of basis functions ω_i, $i=1,2,\cdots,N$ as

$$\sigma(\bar{r}') = \sum_{i=1}^{n} \alpha_i\omega_i(\bar{r}') \qquad (2.12)$$

Thus, the unknown function $\sigma(\bar{r}')$ is tranformed into N unknown constant coeffecients α_i. Substituting Eq. (2.12) into Eq. (2.11), we have

$$\phi(\bar{r})|_S = \sum_{i=1}^{n} \int_S \omega_i(\bar{r}')G(\bar{r},\bar{r}')\ d\bar{r}' \qquad (2.13)$$

Note that the observation point is chosen at the source surface S over which the convolution integral is integrating, because in even-odd mode excitation, the potentials are specified over S.

Next, another set of testing functions t_j, $j=1,2,\ldots,N$ is defined. If the testing functions are chosen exactly the same as the basis functions, the method is named Galerkin's method [11]. Defining an inner product $\langle f,g\rangle = \int_S f(\bar{r})g(\bar{r})d\bar{r}$ for any two real functions f and g, we then take the inner products on both sides of Eq. (2.13) with all the testing functions t_j, and have

$$\int_S \phi(\bar{r})\big|_S t_j(\bar{r})d\bar{r} = \sum_{i=1}^{n} \alpha_i \int_S d\bar{r} \int_S d\bar{r}' \; \omega_i(\bar{r}')G(\bar{r},\bar{r}')t_j(\bar{r})$$
$$j = 1,2,\cdots,N \qquad (2.14)$$

Let us further simplify Eq.(2.14) by defining

$$b_j = \int_S \phi(\bar{r})\big|_S t_j(\bar{r})d\bar{r}$$

and $A_{ij} = \int_S d\bar{r} \int_S d\bar{r}' \omega_i(\bar{r}')G(\bar{r},\bar{r}')t_j(\bar{r})$, $\quad i,j = 1,2,\ldots,N$

Since b_j and A_{ij} are scalars which can be calculated from the known functions $\phi(\bar{r})|_S$, $t_j(\bar{r})$, $\omega_i(\bar{r}')$, and $G(\bar{r},\bar{r}')$, Eq. (2.14) is essentially a system of simultaneous linear algebraic equations. It can be written in the matrix form $Ax = b$. The vector x is the unknown coefficients α_i to be solved. Note that the size of the "kernel" matrix A depends on the number of basis functions. Usually it is considerably smaller than that generated in the finite difference scheme, so a direct method like LU decomposition or Gaussian elimination can be employed. It should also be pointed out that the "kernel" matrix is usually full and those obtained by Galerkin's method should be symmetric.

One of the main tasks in the above procedure is the choice of the basis functions. As mentioned in [11], they should be linearly independent, and the superposition of them as in Eq. (2.12) should approximate the unknown function $\sigma(\bar{r})$ as closely as possible. Some additional factors have to be considered in the choice of both testing functions and basis functions : (1) the accuracy of the solution; (2) the ease of evaluation of the "kernel" matrix's elements, A_{ij}; and (3) the size of the matrix A and its behavior, whether it is well-conditioned or not.

For numerical computation of capacitances using the capacitance model in Sec. 2.1, the basis functions and the testing functions are chosen the same as "pulse" functions which are constant over the subareas on the conductors. In other words, the surfaces of all the conductors are divided into subareas and the charge densities over these subareas are assumed to be constant. Then the above procedure is adopted to solve for the unknown coefficients. The reasons for using pulse functions are as follows : (1) they are linearly independent of each other; (2) the charge density function can be reasonably well expressed in terms of them, provided that the subdivisions are refined enough; (3) a closed-form expression can be derived for the double integral in the evaluation of the kernel matrix elements; (4) if the basis functions and the testing functions are the same, the kernel matrix is not only symmetric but also positive definite, and this guarantees a well-behaved kernel matrix. In order to more closely approximate the charge density function with the choice of pulse-type basis functions, variable width pulses are also used. It improves the accuracy compared with that of constant pulses with the same number of subdivisions. In practice, those variable-width subareas are chosen according to the roots of Chebyshev polynomials [16], because the charges tend to congregate at the edges of the conductors, and Chebyshev polynomials have smaller widths between roots at both ends of the interval [-1,1]. Additionally, an explanation [10] based on expressing the kernel matrix resulting from Galerkin's method as a quadratic form of energy proves that it is a positive definite matrix.

2.4.2. Integral method and even-odd mode analysis

In this section, we show the incorporation of the integral method with the even-odd mode analysis in the case of a multiconductor. Let us continue with the three-conductor example given in Sec. 2.3.2. Assume that the number of subareas on these three conductors are n_1, n_2, n_3, so that the total number of subareas $N = n_1+n_2+n_3$, then Eq. (2.12) can be rewritten as

$$\sigma(\bar{r}') = \sum_{i=1}^{n_1} \alpha_i\omega_i(\bar{r}') + \sum_{i=n_1+1}^{n_1+n_2} \alpha_i\omega_i(\bar{r}') + \sum_{i=n_1+n_2+1}^{N} \alpha_i\omega_i(\bar{r}')$$

Since Galerkin's method is employed, the testing functions are the same as the basis functions, $t_j(\bar{r}) = \omega_j(\bar{r})$, $j=1,2,\cdots,N$, and Eq. (2.14) becomes

$$\int_{S_1} \phi(\bar{r})\Big|_{S_1} \omega_j(\bar{r})\,d\bar{r} = \sum_{i=1}^{n_1} \alpha_i \int_{S_1} d\bar{r} \int_{S_1} d\bar{r}'\,\omega_i(\bar{r}')\,G(\bar{r},\bar{r}')\,\omega_j(\bar{r})$$

$$+ \sum_{i=n_1+1}^{n_1+n_2} \alpha_i \int_{S_1} d\bar{r} \int_{S_2} d\bar{r}'\,\omega_i(\bar{r}')\,G(\bar{r},\bar{r}')\,\omega_j(\bar{r})$$

$$+ \sum_{i=n_1+n_2+1}^{N} \alpha_i \int_{S_1} d\bar{r} \int_{S_3} d\bar{r}'\,\omega_i(\bar{r}')\,G(\bar{r},\bar{r}')\,\omega_j(\bar{r})$$

$$j = 1,2,\ldots,n_1$$

$$\int_{S_2} \phi(\bar{r})\Big|_{S_2} \omega_j(\bar{r})\,d\bar{r} = \sum_{i=1}^{n_1} \alpha_i \int_{S_2} d\bar{r} \int_{S_1} d\bar{r}'\,\omega_i(\bar{r}')\,G(\bar{r},\bar{r}')\,\omega_j(\bar{r})$$

$$+ \sum_{i=n_1+1}^{n_1+n_2} \alpha_i \int_{S_2} d\bar{r} \int_{S_2} d\bar{r}'\,\omega_i(\bar{r}')\,G(\bar{r},\bar{r}')\,\omega_j(\bar{r})$$

$$+ \sum_{i=n_1+n_2+1}^{N} \alpha_i \int_{S_2} d\bar{r} \int_{S_3} d\bar{r}'\,\omega_i(\bar{r}')\,G(\bar{r},\bar{r}')\,\omega_j(\bar{r})$$

$$j = n_1+1,\ldots,n_1+n_2$$

$$\int_{S_3} \phi(\bar{r})\Big|_{S_3} \omega_j(\bar{r})\,d\bar{r} = \sum_{i=1}^{n_1} \alpha_i \int_{S_3} d\bar{r} \int_{S_1} d\bar{r}'\,\omega_i(\bar{r}')\,G(\bar{r},\bar{r}')\,\omega_j(\bar{r})$$

$$+ \sum_{i=n_1+1}^{n_1+n_2} \alpha_i \int_{S_3} d\bar{r} \int_{S_2} d\bar{r}'\,\omega_i(\bar{r}')\,G(\bar{r},\bar{r}')\,\omega_j(\bar{r})$$

$$+ \sum_{i=n_1+n_2+1}^{N} \alpha_i \int_{S_3} d\bar{r} \int_{S_3} d\bar{r}' \omega_i(\bar{r}') G(\bar{r},\bar{r}') \omega_j(\bar{r})$$

$$j = n_1+n_2+1,\ldots,N$$

where S_1, S_2 and S_3 are respectively the areas of the three-conductor surfaces. Note that for different potential patterns, only the potentials at the conductors are different, i.e., only $\phi(\bar{r})|_{S_1}$, $\phi(\bar{r})|_{S_2}$, and $\phi(\bar{r})|_{S_3}$ in the above expressions are changed. This corresponds to only changes in the right-hand side vector b of the matrix equation Ax = b. As an example, for the even and odd modes specified in Sec. 2.3.2, the b vectors are

$$b = \begin{vmatrix} \int_{S_1}\omega_1 \\ \vdots \\ \int_{S_2}\omega_{n_1+1} \\ \vdots \\ \int_{S_3}\omega_{n_1+n_2+1} \\ \vdots \end{vmatrix} \quad \text{for the even mode,}$$

and for the three odd modes,

$$b_1 = \begin{vmatrix} \int_{S_1}\omega_1 \\ \vdots \\ \int_{S_2}\omega_{n_1+1} \\ \vdots \\ -\int_{S_3}\omega_{n_1+n_2+1} \\ -\vdots \end{vmatrix} \qquad b_2 = \begin{vmatrix} \int_{S_1}\omega_1 \\ \vdots \\ -\int_{S_2}\omega_{n_1+1} \\ -\vdots \\ \int_{S_3}\omega_{n_1+n_2+1} \\ \vdots \end{vmatrix} \qquad b_3 = \begin{vmatrix} -\int_{S_1}\omega_1 \\ -\vdots \\ \int_{S_2}\omega_{n_1+1} \\ \vdots \\ \int_{S_3}\omega_{n_1+n_2+1} \\ \vdots \end{vmatrix}$$

On the other hand, the kernel matrix A only concerns the Green's function and the basis functions, and it stays the same for different even-odd mode excitations. This is one of the important advantages of the integral method. Because the evaluation of the kernel matrix elements A_{ij} always is the most computationally intensive, we have to compute it only once, and LU factorize it only once, then use backward substitution to find the charge densities corresponding to b_1, b_2, and b_3 as listed above. If the differential method is used, we have to essentially solve four boundary-value problems, and cannot take advantage of the even-odd mode excitations as above. From this point of view, the integral method is better in the case of multiple conductors.

2.5. Derivation of Two-Dimensional Green's Function

Two-dimensional Green's functions to Poisson's equation in a homogeneous medium or in two layers of media have been derived in [4,14], and they are given here for reference; on the other hand, we will concentrate on the derivation of the three-layer case in this section.

2.5.1. Homogeneous medium

Consider a unit line charge at (x',y') in a homogeneous medium with dielectric constant ε, and the ground plane is at $y = 0$. Then the Green's function is

$$G(x,y|x',y') = \frac{-1}{2\pi\varepsilon} \ln \frac{(x-x')^2+(y-y')^2}{(x-x')^2+(y+y')^2} \qquad (2.15)$$

Note that the logarithmic function in the above expression characterizes a two-dimensional line source, and the complementary terms of $y-y'$ and $y+y'$ reflect the consequence of the source and the image across the ground plane.

2.5.2. Two layers of media

Let there be two layers of media over the ground plane as shown in Fig. 2.4. The first layer has a thickness h over the ground plane and a dielectric constant ε_1. The second layer on top of the first one has a dielectric constant ε_2. There are four subfunctions associated with the Green's function depending on the relative positions of source point (x',y') and observation point (x,y) namely: $G_{11}(x,y|x',y')$, $G_{21}(x,y|x',y')$, $G_{22}(x,y|x',y')$ and $G_{12}(x,y|x',y')$. The first digit of the subscript specifies the region where the observation point resides, and the second digit indicates the position of the source point. This convention will also be used for other subfunctions of other Green's functions discussed later. These four subfunctions have been derived by the method of multiple reflections across the ground plane and the interface between the two media [4]. They are listed as :

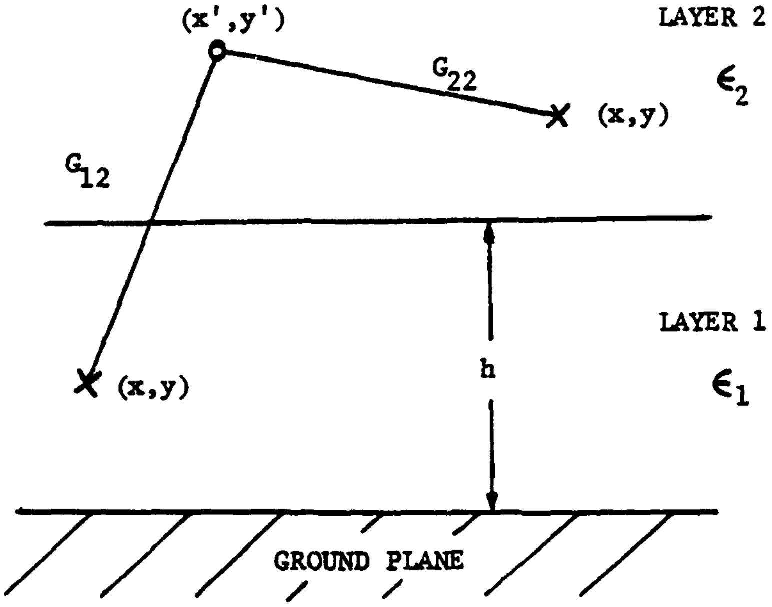


Figure 2.4 Derivation of two-dimensional Green's function in two layers of media.

$$G_{11}(x,y|x',y') = \frac{1}{4\pi\varepsilon_1}\sum_{k=0}^{\infty}\left(\frac{\varepsilon_2-\varepsilon_1}{\varepsilon_2+\varepsilon_1}\right)^k \ln\frac{(x-x')^2+(y+y'+2kh)^2}{(x-x')^2+(y-y'+2kh)^2}$$

$$+\frac{1}{4\pi\varepsilon_1}\sum_{k=0}^{\infty}\left(\frac{\varepsilon_2-\varepsilon_1}{\varepsilon_2+\varepsilon_1}\right)^{k+1} \ln\frac{(x-x')^2+(y+y'-2(k+1)h)^2}{(x-x')^2+(y-y'-2(k+1)h)^2}$$

$0 \leq y \leq h$ and $0 \leq y' \leq h$ (2.16a)

$$G_{22}(x,y|x',y') = \frac{1}{4\pi\varepsilon_2}\sum_{k=0}^{\infty}\left(\frac{\varepsilon_2-\varepsilon_1}{\varepsilon_2+\varepsilon_1}\right)^k \ln\frac{(x-x')^2+(y+y'+2kh)^2}{(x-x')^2+(y-y'-2kh)^2}$$

$$-\frac{1}{4\pi\varepsilon_2}\sum_{k=0}^{\infty}\left(\frac{\varepsilon_2-\varepsilon_1}{\varepsilon_2+\varepsilon_1}\right)^{k+1} \ln\frac{(x-x')^2+(y+y'+2(k-1)h)^2}{(x-x')^2+(y-y'-2(k-1)h)^2}$$

$0 < h \leq y$ and $0 < h \leq y'$ (2.16b)

$$G_{12}(x,y|x',y') = \frac{1}{2\pi(\varepsilon_2+\varepsilon_1)}\sum_{k=0}^{\infty}\left(\frac{\varepsilon_2-\varepsilon_1}{\varepsilon_2+\varepsilon_1}\right)^k \ln\frac{(x-x')^2+(y+y'+2kh)^2}{(x-x')^2+(|y-y'|+2kh)^2}$$

$0 \leq y \leq h \leq y'$ (2.16c)

$$G_{21}(x,y|x',y') = G_{12}(x,y|x',y') \text{ except}$$

$0 \leq y' \leq h \leq y$ (2.16d)

Compared with the formula in Eq. (2.15), the above formulas basically are still logarithmic-type functions with some modification due to the two media. The infinite summation in the formula is a direct result of the infinite reflections of images between the two infinite parallel planes.

2.5.3. Three layers of media

The method of multiple reflections employed in [4] to derive the Green's function for two layers of media is too complicated to formulate the three-layer case. Hence, another method of boundary matching in the spectral domain [19] is employed instead.

As shown in Fig. 2.5, the dielectric constants of the three layers of media are ε_1, ε_2, and ε_0. The outermost layer is assumed to be free space always, and the interconnect will not reside in this region. Thus, there are four subfunctions, G_{11}, G_{21}, and G_{12}, G_{22}, to be considered. The convention on the subscript of those subfunctions is the same as that discussed in the previous section. Let us consider the case when source point (x',y') is in the layer ε_2, and derive the subfunctions $G_{22}(x,y|x',y')$ and $G_{12}(x,y|x',y')$. The other two subfunctions can be derived similarly.

As depicted in Fig. 2.5, an artificial interface is drawn past the source point (x',y') in layer ε_2, which increases the number of regions to four. In analogy to Fourier analysis, assume that the potentials in these four regions are :

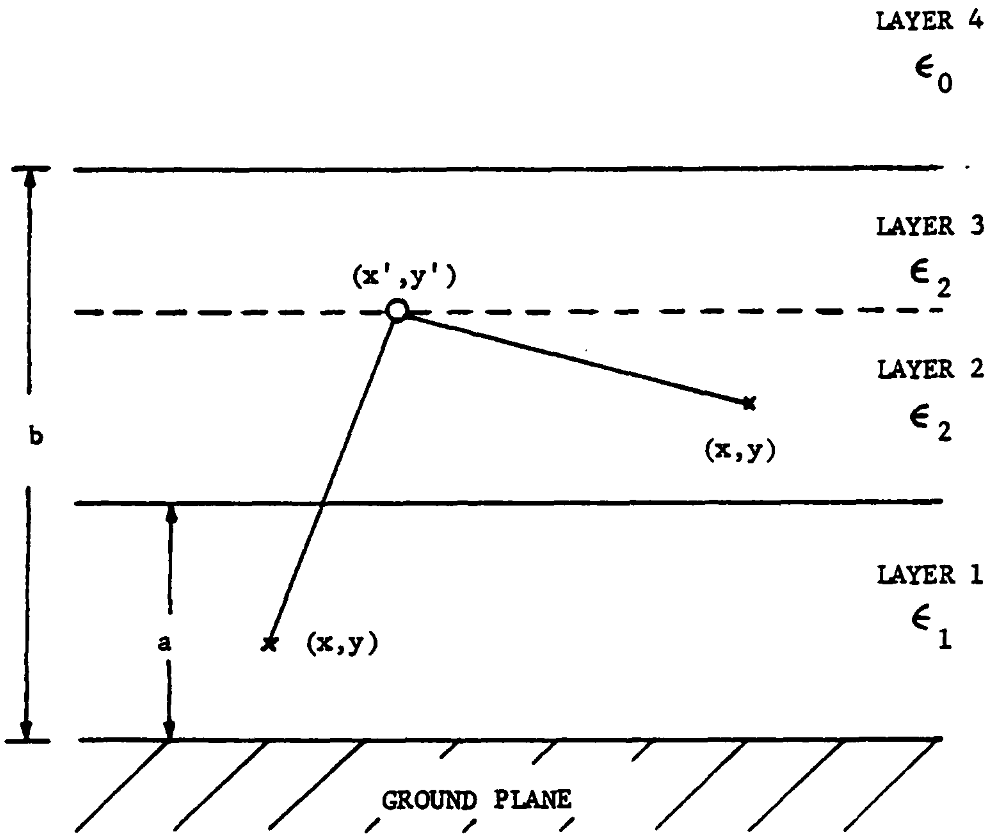


Figure 2.5 Derivation of two-dimensional Green's function in three layers of media.

$$\phi_1(x,y) = \int_0^\infty \cos(k(x-x'))[\alpha_1(k)\sinh(ky)]dk \qquad (2.17a)$$

$$\phi_2(x,y) = \int_0^\infty \cos(k(x-x'))[\alpha_2(k)\sinh(ky) + \beta_2(k)\cosh(ky)]dk \qquad (2.17b)$$

$$\phi_3(x,y) = \int_0^\infty \cos(k(x-x'))[\alpha_3(k)\sinh(ky) + \beta_3(k)\cosh(ky)]dk \qquad (2.17c)$$

$$\phi_4(x,y) = \int_0^\infty \cos(k(x-x'))[\alpha_4(k)e^{-ky}]dk \qquad (2.17d)$$

Note that the $\alpha_i(k)$'s and $\beta_i(k)$'s in Eq. (2.17) may be considered as the unknown spectral components of the corresponding potentials. Because of the ground plane in region 1, potential $\phi_1(x,y)$ should be zero at $y = 0$ and does not contain a cosh(ky) term. For the outermost region 4, e^{-ky} represents a decaying wave in the open boundary.

At the interfaces between different regions, the potentials and the corresponding "displacement vectors" $\bar{D}$'s have to be continuous. Hence, the boundary conditions can be written as follows :

$$\phi_1|_{y=a} = \phi_2|_{y=a} \qquad (2.18a)$$

$$\varepsilon_1 \frac{\partial \phi_1}{\partial y}\Big|_{y=a} = \varepsilon_2 \frac{\partial \phi_2}{\partial y}\Big|_{y=a} \qquad (2.18b)$$

$$\phi_2|_{y=y'} = \phi_3|_{y=y'} \tag{2.18c}$$

$$\varepsilon_2 \frac{\partial \phi_3}{\partial y}\Big|_{y=y'} - \varepsilon_2 \frac{\partial \phi_2}{\partial y}\Big|_{y=y'} = \delta(y-y') \tag{2.18d}$$

$$\phi_3|_{y=b} = \phi_4|_{y=b} \tag{2.18e}$$

$$\varepsilon_2 \frac{\partial \phi_3}{\partial y}\Big|_{y=b} = \varepsilon_0 \frac{\partial \phi_4}{\partial y}\Big|_{y=b} \tag{2.18f}$$

Note that these six boundary conditions uniquely determine the six unknown spectral components, $\alpha_i(k)$'s and $\beta_i(k)$'s in Eq. (2.17). After some tedious algebraic manipulation, the spectral components can be solved :

$$\alpha_1(k) = \frac{2}{\pi(\varepsilon_1+\varepsilon_2)} \frac{1}{kD(k)} e^{-ky'}(1-\xi_2 e^{-2k(b-y')}) \tag{2.19a}$$

$$\alpha_2(k) = \frac{1}{\pi\varepsilon_2} \frac{1}{kD(k)} e^{-ky'}(1-\xi_1\cosh(2ka))(1-\xi_2 e^{-2k(b-y')}) \tag{2.19b}$$

$$\beta_2(k) = \frac{1}{\pi\varepsilon_2} \frac{1}{kD(k)} e^{-ky'}(\xi_1\sinh(2ka))(1-\xi_2 e^{-2k(b-y')}) \tag{2.19c}$$

$$\alpha_3(k) = \frac{-1}{\pi\varepsilon_2}\,\frac{1}{kD(k)}(\sinh(ky')+\xi_1\sinh(k(2a-y')))(1+\xi_2 e^{-2kb}) \tag{2.19d}$$

$$\beta_3(k) = \frac{1}{\pi\varepsilon_2}\,\frac{1}{kD(k)}(\sinh(ky')+\xi_1\sinh(k(2a-y')))(1-\xi_2 e^{-2kb}) \tag{2.19e}$$

$$\alpha_4(k) = \frac{2}{\pi(\varepsilon_1+\varepsilon_2)}\,\frac{1}{kD(k)}(\sinh(ky')+\xi_1\sinh(k(2a-y'))) \tag{2.19f}$$

where $\xi_1 = \dfrac{\varepsilon_2-\varepsilon_1}{\varepsilon_2+\varepsilon_1}$ and $\xi_2 = \dfrac{\varepsilon_0-\varepsilon_2}{\varepsilon_0+\varepsilon_2}$

and $D(k) = 1 - \xi_1 e^{-2ka} - \xi_2 e^{-2kb} + \xi_1\xi_2 e^{-2k(b-a)}$

Substituting Eq. (2.19) back into Eq. (2.17), we can obtain the potentials in spectral integral form analogous to Fourier integrals. Since the source in the above derivation is in the layer ε_2, the potentials $\emptyset_2$ and $\emptyset_3$, both in layer ε_2 as a result of the artificial interface introduced in the derivation, can be combined to form the subfunction G_{22}. $\emptyset_1$ is naturally the subfunction G_{12}.

Written in spectral integral form, $\emptyset_2$, as an example, can be expressed as

$$\emptyset_2(x,y) = \frac{1}{2\pi\varepsilon_2}\int_0^\infty \frac{\cos(k(x-x'))}{kD(k)}(1-\xi_2 e^{-2k(b-y')}).$$

$$(e^{-k(y'-y)}-e^{-k(y'+y)}+\xi_1 e^{-k(y'+y-2a)}-\xi_1 e^{-k(y'-y+2a)})dk$$

$$b > y' \geq y \geq a \qquad (2.20)$$

In order to "reverse Fourier transform" Eq. (2.20), or integrate out the dummy variable k, first the factor $[D(k)]^{-1}$ is expanded in binomial form as :

$$[D(k)]^{-1} = \sum_{j=0}^{\infty} \sum_{n=0}^{j} \sum_{k=0}^{n} (-1)^k \binom{j}{n} \binom{n}{k} \xi_1^{j-n+k} \xi_2^n e^{-2kbn} e^{-2ka(j-n-k)}$$

Then substituting this binomial expansion into Eq. (2.20) and utilizing the following definite integral from [17] :

$$\int_0^{\infty} \frac{e^{-\gamma x}-e^{-\beta x}}{x} \cos(bx)\, dx = (1/2) \ln \frac{b^2+\beta^2}{b^2+\gamma^2} \qquad \mathrm{Re}\beta>0,\ \mathrm{Re}\gamma\geq 0$$

we can integrate out the spectral variable k and express $\phi_2(x,y)$ in the space domain as :

$$\phi_2(x,y) = \frac{1}{4\pi\varepsilon_2} \sum_{j=0}^{\infty} \sum_{n=0}^{j} \sum_{k=0}^{n} (-1)^k \binom{j}{n} \binom{n}{k} \xi_1^{j-n+k} \xi_2^n$$

$$\left[\ln \frac{(x-x')^2+(y+y'+2bn+2a(j-k-n))^2}{(x-x')^2+(-y+y'+2bn+2a(j-k-n))^2} \right.$$

$$+ \xi_1 \ln \frac{(x-x')^2+(-y+y'+2bn+2a(j-k-n+1))^2}{(x-x')^2+(y+y'+2bn+2a(j-k-n-1))^2}$$

$$+ \xi_2 \ln \frac{(x-x')^2+(-y-y'+2b(n+1)+2a(j-k-n))^2}{(x-x')^2+(y-y'+2b(n+1)+2a(j-k-n))^2}$$

$$+ \xi_1\xi_2 \ln \frac{(x-x')^2+(y-y'+2b(n+1)+2a(j-k-n-1))^2}{(x-x')^2+(-y-y'+2b(n+1)+2a(j-k-n+1))^2} \Bigg]$$

$$0 < a \le y \le y' \le b \qquad (2.21)$$

Because the constraint on $\emptyset_3(x,y)$ is $a \le y' \le y < b$, and $\emptyset_3$ has a similar formula as that in Eq. (2.21), the combination of $\emptyset_2$ and $\emptyset_3$ results in substituting absolute values for $y-y'$ in the above Eq. (2.21). The other subfunctions can be obtained in a similar manner. In summary, we have the two-dimensional Green's function in three layers of media as follows :

$$G_{11}(x,y|x',y') = \frac{1}{4\pi\varepsilon_1} \sum_{j=0}^{\infty} \sum_{n=0}^{j} \sum_{k=0}^{n} (-1)^k \binom{j}{n}\binom{n}{k} \xi_1{}^{j-n+k} \xi_2{}^n$$

$$\Bigg[\ln \frac{(x-x')^2+(|y+y'|+2bn+2a(j-k-n))^2}{(x-x')^2+(|y-y'|+2bn+2a(j-k-n))^2}$$

$$+ \xi_1 \ln \frac{(x-x')^2+(-|y+y'|+2bn+2a(j-k-n+1))^2}{(x-x')^2+(-|y-y'|+2bn+2a(j-k-n+1))^2}$$

$$+ \xi_2 \ln \frac{(x-x')^2+(-|y+y'|+2b(n+1)+2a(j-k-n))^2}{(x-x')^2+(-|y-y'|+2b(n+1)+2a(j-k-n))^2}$$

$$+ \xi_1\xi_2 \ln \frac{(x-x')^2+(|y+y'|+2b(n+1)+2a(j-k-n-1))^2}{(x-x')^2+(|y-y'|+2b(n+1)+2a(j-k-n-1))^2}\Bigg]$$

$$0 \le y,y' \le a < b \qquad (2.22a)$$

$$G_{21}(x,y|x',y') = G_{12}(x,y|x',y') \text{ except} \qquad (2.22b)$$

$$0 \le y' \le a \le y \le b$$

$$G_{22}(x,y|x',y') = \frac{1}{4\pi\varepsilon_2} \sum_{j=0}^{\infty} \sum_{n=0}^{j} \sum_{k=0}^{n} (-1)^k \binom{j}{n}\binom{n}{k} \xi_1^{\,j-n+k} \xi_2^{\,n}$$

$$\Bigg[\ln \frac{(x-x')^2+(|y+y'|+2bn+2a(j-k-n))^2}{(x-x')^2+(|y-y'|+2bn+2a(j-k-n))^2}$$

$$+ \xi_1 \ln \frac{(x-x')^2+(|y-y'|+2bn+2a(j-k-n+1))^2}{(x-x')^2+(|y+y'|+2bn+2a(j-k-n-1))^2}$$

$$+ \xi_2 \ln \frac{(x-x')^2+(-|y+y'|+2b(n+1)+2a(j-k-n))^2}{(x-x')^2+(-|y-y'|+2b(n+1)+2a(j-k-n))^2}$$

$$+\ \xi_1\xi_2\ \ln \frac{(x-x')^2+(-|y-y'|+2b(n+1)+2a(j-k-n-1))^2}{(x-x')^2+(-|y+y'|+2b(n+1)+2a(j-k-n+1))^2}\Bigg]$$

$$0 < a \le y,y' \le b \qquad (2.22c)$$

$$G_{12}(x,y|x',y') = \frac{1}{2\pi(\varepsilon_1+\varepsilon_2)} \sum_{j=0}^{\infty} \sum_{n=0}^{j} \sum_{k=0}^{n} (-1)^k \binom{j}{n}\binom{n}{k} \xi_1^{\,j-n+k}\xi_2^{\,n}$$

$$\Bigg[\ \ln \frac{(x-x')^2+(|y+y'|+2bn+2a(j-k-n))^2}{(x-x')^2+(|y-y'|+2bn+2a(j-k-n))^2}$$

$$+\ \xi_2\ \ln \frac{(x-x')^2+(-|y+y'|+2b(n+1)+2a(j-k-n))^2}{(x-x')^2+(-|y-y'|+2b(n+1)+2a(j-k-n))^2}\Bigg]$$

$$0 \le y \le a \le y' \le b \qquad (2.22d)$$

Note that these expressions still retain the basic logarithmic function as that in Eq. (2.15). With the pulse basis function, the double integral in the evaluation of the kernel elements will result in a closed form. The detailed expressions of these closed-form evaluations are included in Appendix C. Furthermore, the complex triple summation in the above formulas can be interpreted as the multiple reflections among the three parallel planes, i.e., the ground plane and two dielectric interfaces. The program which executes the aforementioned procedure of the integral method is named "CAP2D" and a user's manual for it is included in Appendix B.

2.6. Derivation of Three-Dimensional Green's Function

There are some circumstances in which the two-dimensional analysis cannot satisfactorily compute the interconnect capacitances. For example, the cross-over of two perpendicular interconnects on different levels is one such example. Hence, the three-dimensional Green's functions in homogeneous, two-layer and three-layer media are examined in this section.

2.6.1. Homogeneous medium

Consider a unit point charge at $\bar{r}' = (x',y',z')$ in a homogeneous medium with dielectric constant ε and the ground plane at $z = 0$. The Green's function to the Poisson's equation satisfying this boundary condition has been derived [6] and is written as :

$$G(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon}\{[\rho^2+(z-z')^2]^{-1/2} - [\rho^2+(z+z')^2]^{-1/2}\} \qquad (2.23)$$

where $\rho^2 = (x-x')^2+(y-y')^2$

Note that $\bar{r} = (x,y,z)$ and $\bar{r}' = (x',y',z')$ are the observation and the source point, respectively. The core of the expression is a function proportional to the reciprocal of the distance which is different from the logarithmic function in the two-dimensional case.

2.6.2. Two layers of media

The basic principle used in the derivation of the three-dimensional Green's function in multiple layers of media is similar to boundary matching in the spectral domain as discussed in Sec. 2.5 for the two-dimensional case. The formulation had been introduced in [6]. For two layers of media, let the dielectric constant of the lower layer be ε_1 and the thickness be h. Assume that the ground plane is at $z = 0$, and the dielectric constant of the top layer is ε_2. Utilizing the same convention as the two-dimensional Green's function, the three-dimensional Green's function for two layers of media is

$$G_{11}(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon_1 r} - \frac{1}{4\pi\varepsilon_1}\sum_{n=0}^{\infty}\xi^n\Big[[\rho^2+(z+z'+2nh)^2]^{-1/2}$$

$$- \xi[\rho^2+(z-z'+2(n+1)h)^2]^{-1/2} - \xi[\rho^2+(z-z'-2(n+1)h)^2]^{-1/2}$$

$$+ \xi[\rho^2+(z+z'-2(n+1)h)^2]^{-1/2}\Big]$$

$$0 < z,z' \le h \qquad (2.24a)$$

$$G_{22}(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon_2 r} - \frac{1}{4\pi\varepsilon_2}\sum_{n=0}^{\infty}\xi^n\Big[-\xi[\rho^2+(z+z'+2(n-1)h)^2]^{-1/2}$$

$$+ [\rho^2+(z+z'+2nh)^2]^{-1/2}\Big]$$

$$0 < h \leq z,z' \tag{2.24b}$$

$$G_{12}(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon_2}\left(\frac{2\varepsilon_2}{\varepsilon_2+\varepsilon_1}\right) \sum_{n=0}^{\infty} \xi^n\Big[[\rho^2+(z-z'-2nh)^2]^{-1/2}$$

$$- [\rho^2+(z+z'+2nh)^2]^{-1/2}\Big]$$

$$0 < z \leq h \leq z' \tag{2.24c}$$

$$G_{21}(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon_1}\left(\frac{2\varepsilon_1}{\varepsilon_2+\varepsilon_1}\right) \sum_{n=0}^{\infty} \xi^n\Big[[\rho^2+(z-z'+2nh)^2]^{-1/2}$$

$$- [\rho^2+(z+z'+2nh)^2]^{-1/2}\Big]$$

$$0 < z' \leq h \leq z \tag{2.24d}$$

where $\rho^2 = (x-x')^2+(y-y')^2$, $r^2 = \rho^2+(z-z')^2$ and $\xi = \dfrac{\varepsilon_2-\varepsilon_1}{\varepsilon_2+\varepsilon_1}$

2.6.3. Three layers of media

Consider the configuration in Fig. 2.6. The thicknesses of the layers are a and b-a and the dielectric constants are ε_0, ε_2, and ε_1. These parameters are the same as those in the two-dimensional case.

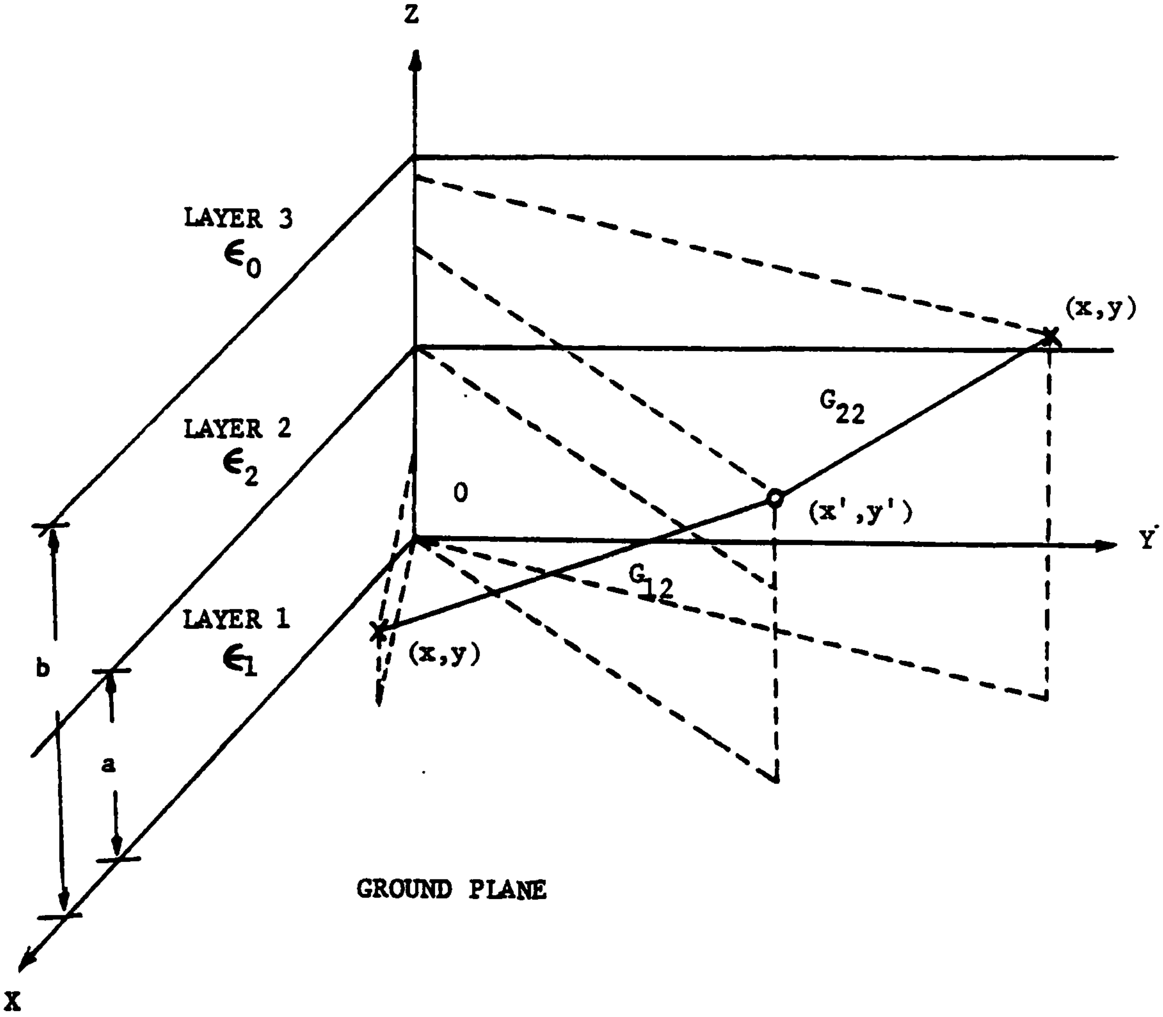


Figure 2.6 Derivation of three-dimensional Green's function in three layers of media.

Utilizing one well-known Fourier integral [15,6]:

$$r^{-1} = \int_0^\infty J_0(k\rho)\; e^{-k|z-z'|}\, dk \qquad (2.25)$$

where $r^2 = \rho^2 + (z-z')^2$, and $J_0(x)$ is the Bessel's function of first kind order 0, we can derive the Green's function.

First, consider the case in which the source point resides in layer ε_2 and the subfunctions G_{12} and G_{22} are derived. As before, the potentials in layers ε_0, ε_2, ε_1, are represented as Fourier integrals:

$$\phi_0 = \frac{1}{4\pi\varepsilon_0}\int_0^\infty \Omega(k) J_0(k\rho) e^{-k(z-z')}\, dk$$

$$0 < a \le z' \le b \le z \qquad (2.26a)$$

$$\begin{aligned}\phi_2 = G_{22}(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon_0}\Big[&\int_0^\infty \Big(\frac{\varepsilon_0}{\varepsilon_2}\Big) J_0(k\rho) e^{-k|z-z'|} dk \\ &+ \int_0^\infty \theta_1(k) J_0(k\rho) e^{-k(z-z')} dk \\ &+ \int_0^\infty \theta_2(k) J_0(k\rho) e^{k(z-z')} dk\Big]\end{aligned}$$

$$0 < a \le z,\ z' \le b \qquad (2.26b)$$

$$\phi_1 = G_{12}(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon_0}\Big[\int_0^\infty \tau_1(k) J_0(k\rho) e^{-k(z-z')} dk$$

$$+ \int_0^\infty \tau_2(k) J_0(k\rho) e^{k(z-z')} dk \Big]$$

$$0 \le z \le a \text{ and } a \le z' \le b \qquad (2.26c)$$

where $\rho^2 = (x-x')^2 + (y-y')^2$

The functions $\Omega(k)$, $\Theta_1(k)$, $\Theta_2(k)$, $\tau_1(k)$, and $\tau_2(k)$ are the unknown spectral components to be determined by boundary matching. As in the two-dimensional case, the exponential term $e^{-k(z-z')}$ in $\emptyset_0$ dictates a decaying wave towards the open boundary in layer ε_0; also, the absolute value on z-z' is removed because z is always greater than z' in this region. Note that the first term in $\emptyset_2$ accounts for the singularity of the source point in the same region. Because the relative magnitudes of z and z' are not fixed in this region, the absolute value is needed in the exponent.

As in the two-dimensional case, the continuity of the potentials and the displacement vectors dictates the following necessary boundary conditions :

$$\emptyset_1|_{z=0} = 0 \qquad (2.27a)$$

$$\emptyset_1|_{z=a} = \emptyset_2|_{z=a} \qquad (2.27b)$$

$$\varepsilon_1 \frac{\partial \emptyset_1}{\partial z}\Big|_{z=a} = \varepsilon_2 \frac{\partial \emptyset_2}{\partial z}\Big|_{z=a} \qquad (2.27c)$$

$$\phi_2|_{z=b} = \phi_0|_{z=b} \qquad (2.27d)$$

$$\varepsilon_2\frac{\partial\phi_2}{\partial z}\Big|_{z=b} = \varepsilon_0\frac{\partial\phi_0}{\partial z}\Big|_{z=b} \qquad (2.27e)$$

Substituting Eq. (2.26) into Eq. (2.27), we can solve the five unknown spectral components in Eq. (2.26). After some algebraic manipulation, they are listed as follows :

$$\theta_1(k) = -\frac{\varepsilon_0}{\varepsilon_2}\Delta^{-1}(\xi_1-e^{-2ka})(\xi_2-e^{2k(b-z')}) \qquad (2.28a)$$

$$\theta_2(k) = -\xi_2\frac{\varepsilon_0}{\varepsilon_2}\Delta^{-1}[e^{-2k(a-z')}(1-\xi_1 e^{-2ka}) + (\xi_1-e^{-2ka})] \qquad (2.28b)$$

$$\tau_1(k) = \frac{2\varepsilon_0}{\varepsilon_1+\varepsilon_2}\Delta^{-1}e^{-2ka}(\xi_1-e^{2k(b-z')}) \qquad (2.28c)$$

$$\tau_2(k) = \frac{2\varepsilon_0}{\varepsilon_1+\varepsilon_2}\Delta^{-1}e^{-2ka}(e^{2kb}-\xi_1 e^{2kz'}) \qquad (2.28d)$$

where $\Delta = \xi_1\xi_2 - \xi_2 e^{-2ka} - \xi_1 e^{2k(b-2a)} + e^{2k(b-a)}$

and $\xi_1 = \dfrac{\varepsilon_2-\varepsilon_1}{\varepsilon_2+\varepsilon_1}$, $\quad \xi_2 = \dfrac{\varepsilon_0-\varepsilon_2}{\varepsilon_0+\varepsilon_2}$

Note that the function $\Omega(k)$ is not associated with the Green's function in question, so it is not listed above. Utilizing the binomial expansion of Δ^{-1} and the following definite integrals [17] :

$$\int_0^\infty e^{-\alpha x} J_0(\beta x)\,dx = (\alpha^2 + \beta^2)^{-1/2}, \qquad \mathrm{Re}(\alpha + i\beta) > 0$$

$$\int_0^\infty e^{-\alpha x} \cosh(\beta x) J_0(\gamma x) = 0.5\{[\gamma^2 + (\beta-\alpha)^2]^{-1/2} + [\gamma^2 + (\beta-\alpha)^2]^{-1/2}\},$$
$$\mathrm{Re}\alpha > |\mathrm{Re}\beta|, \ \gamma > 0$$

we can transform the Fourier integrals in Eq. (2.26) back into the space domain in order to obtain the two subfunctions of the three-dimensional Green's function in three layers of media as follows :

$$G_{22}(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon_2 r} - \frac{1}{4\pi\varepsilon_2} \sum_{j=0}^{\infty} \sum_{n=0}^{j} \sum_{k=0}^{n} (-1)^{n-k} \binom{j}{n} \binom{n}{k} \xi_1{}^{n} \xi_2{}^{j-k}.$$

$$\begin{aligned}
&\Big[[\rho^2 + (2((j-k)b + (2k-n)a) + z + z')^2]^{-1/2} \\
&- \xi_1[\rho^2 + (2((j-k)b + (2k-n-1)a) + z + z')^2]^{-1/2} \\
&- \xi_2\{[\rho^2 + (2((j-k+1)b + (2k-n)a) - z + z'))^2]^{-1/2} \\
&\quad + [\rho^2 + (2((j-k+1)b + (2k-n)a) + z - z'))^2]^{-1/2} \\
&\quad - [\rho^2 + (2((j-k+1)b + (2k-n)a) - z - z'))^2]^{-1/2}\} \\
&+ \xi_1\xi_2\{[\rho^2 + (2((j-k+1)b + (2k-n-1)a) - z + z'))^2]^{-1/2}
\end{aligned}$$

$$+ \;[\rho^2+(2((j-k+1)b+(2k-n-1)a)+z-z'))^2]^{-1/2}$$

$$- \;[\rho^2+(2((j-k+1)b+(2k-n+1)a)-z-z'))^2]^{-1/2}\}\Big]$$

$$0 < a \le z \le b \text{ and } 0 < a \le z' \le b \qquad (2.29a)$$

$$G_{12}(\bar{r},\bar{r}') = \frac{1}{2\pi(\varepsilon_1+\varepsilon_2)} \sum_{j=0}^{\infty} \sum_{n=0}^{j} \sum_{k=0}^{n} (-1)^{n-k}\binom{j}{n}\binom{n}{k}\xi_1{}^{n}\xi_2{}^{j-k}.$$

$$\Big[\{[\rho^2+(2((j-k)b+(2k-n)a)+|z-z'|))^2]^{-1/2}$$

$$- \;[\rho^2+(2((j-k)b+(2k-n)a)+|z+z'|))^2]^{-1/2}\}$$

$$- \;\xi_2\{[\rho^2+(2((j-k+1)b+(2k-n)a)-|z+z'|))^2]^{-1/2}$$

$$+ \;[\rho^2+(2((j-k+1)b+(2k-n)a)-|z-z'|))^2]^{-1/2}\}\Big]$$

$$0 \le z \le a < b \text{ and } 0 < a \le z' \le b \qquad (2.29b)$$

The other two subfunctions of the Green's function can be similarly formulated and they are

$$G_{11}(\bar{r},\bar{r}') = \frac{1}{4\pi\varepsilon_1 r} - \frac{1}{4\pi\varepsilon_1} \sum_{j=0}^{\infty} \sum_{n=0}^{j} \sum_{k=0}^{n} (-1)^{k}\binom{j}{n}\binom{n}{k}\xi_1{}^{n}\xi_2{}^{j-n+k}.$$

$$\Big[[\rho^2+(2(-(j-k)z'+(j-n+k)b+(n-2k)a)+z+z'))^2]^{-1/2}$$

$$+ \xi_1\xi_2[\rho^2+(2(-(j-k)z'+(j-n+k+1)b+(n-2k-1)a)+z+z'))^2]^{-1/2}$$

$$- \xi_1\{[\rho^2+(2(-(j-k)z'+(j-n+k)b+(n-2k+1)a)+z-z'))^2]^{-1/2}$$

$$- [\rho^2+(2(-(j-k)z'+(j-n+k)b+(n-2k+1)a)-z-z'))^2]^{-1/2}$$

$$+ [\rho^2+(2(-(j-k)z'+(j-n+k)b+(n-2k+1)a)-z+z'))^2]^{-1/2}\}$$

$$- \xi_2\{[\rho^2+(2(-(j-k)z'+(j-n+k+1)b+(n-2k)a)+z-z'))^2]^{-1/2}$$

$$- [\rho^2+(2(-(j-k)z'+(j-n+k+1)b+(n-2k)a)-z-z'))^2]^{-1/2}$$

$$+ [\rho^2+(2(-(j-k)z'+(j-n+k+1)b+(n-2k)a)-z+z'))^2]^{-1/2}\}\Big]$$

$$0 \leq z \leq a < b \text{ and } 0 \leq z' \leq a < b \qquad (2.29c)$$

$$G_{21}(\bar{r},\bar{r}') = G_{12}(\bar{r},\bar{r}') \quad \text{except}$$

$$0 \leq z' \leq a < b \text{ and } 0 < a \leq z \leq b \qquad (2.29d)$$

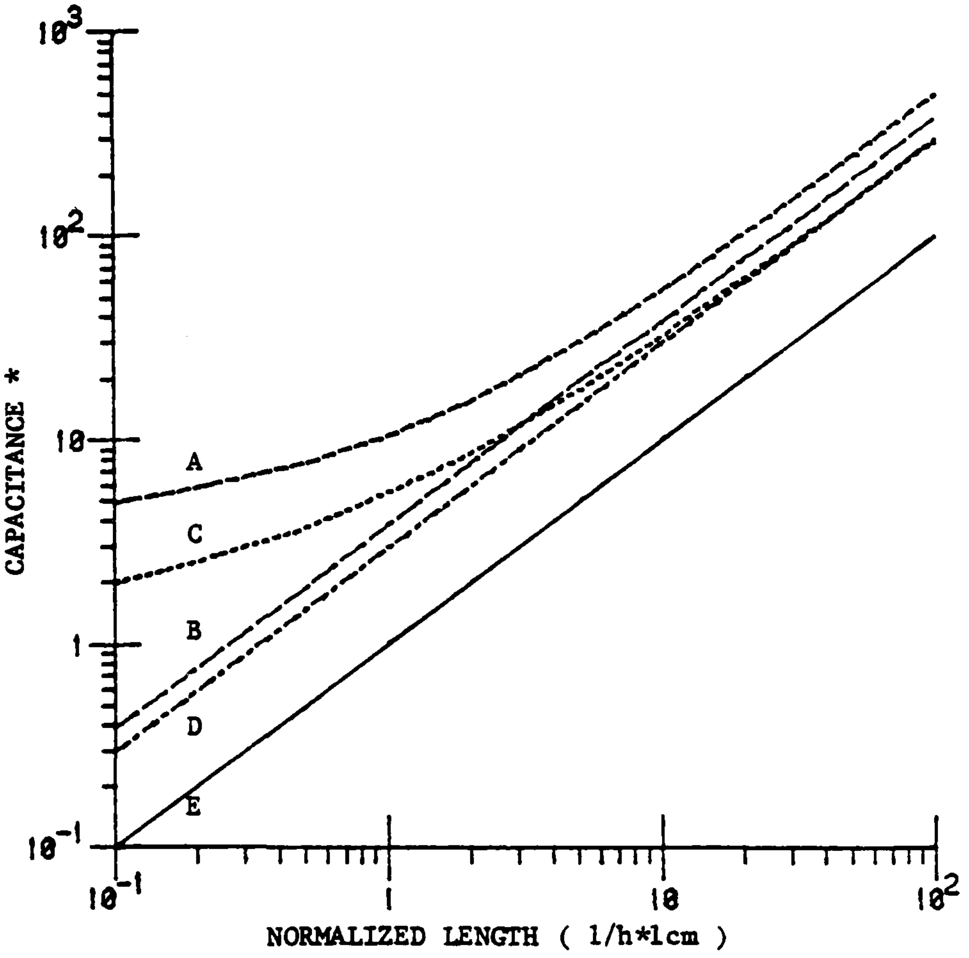
It should be pointed out that the "core" function in the above expressions is still the inverse square root function mentioned earlier. Associated with the pulse basis functions in Appendix D, the inverse square root function results in closed-form integration for the evaluation of the kernel matrix elements. A detailed derivation of these kernel integrals is included in Appendix D. The program which implements the numerical computation of the three-dimensional interconnect capacitances described above is named "CAP3D", and the user's manual for it is the same as the one for "CAP2D", and is included in Appendix B with some examples.

2.7. Some Results of Two- and Three-Dimensional Numerical Computation

2.7.1. Comparison of two- and three-dimensional capacitance

In two-dimensional capacitance computation, the length of the interconnect is assumed to be infinite; in other words, the calculated capacitance is the capacitance per unit length of the interconnect. In comparison with the three-dimensional capacitance, as shown in Fig. 2.7, the capacitances are plotted versus the "normalized" length. Since the two-dimensional capacitances and the parallel-plate capacitance are proportional to the length, they are linear lines in Fig. 2.7. As expected, the discrepancies between the corresponding three-dimensional and two-dimensional capacitances are diminishing as the length increases. Some of the difference may be attributed to the "end effect" or the fringing-field capacitances at the ends of the interconnect which are included in the three-dimensional computation. Also, the difference between two-dimensional capacitance and the parallel-plate capacitance is due to the fringing-field capacitance at both sides of the interconnect which had been taken into account in the two-dimensional computation.

Two cases of thickness are also considered in Fig. 2.7. In the case of $t/h = 0$, the interconnect is considered infinitesimally thin,



A : 3-d and t/h = 0.8
B : 2-d and t/h = 0.8
C : 3-d and t/h = 0
D : 2-d and t/h = 0
E : parallel-plate capacitance

* Capacitances are in units of $8.854 \times 10^{-14} \varepsilon$ Farad, if the length is in units of centimeter, where ε is the dielectric constant of the homogeneous medium.

Figure 2.7 Comparison of two- and three-dimensional capacitance (w/h = 1).

and, naturally, the fringing-field capacitances from the side walls are reduced and all the corresponding capacitances are smaller compared with those of thickness t/h = 0.8. From this comparison, it can be concluded that as the dimensions of interconnects are reduced, the fringing-field capacitance does not decrease in the same proportion as the parallel-plate field capacitance. Thus, the fringing capacitance gradually emerges as the dominating component in the evaluation of the capacitances. Hence, more accurate computation is needed to evaluate them in order to take all the capacitance components into account.

2.7.2. Coupling capacitance and slant side walls

Shown in Fig. 2.8 are the cross-sections of two interconnects. The self- and coupling capacitances with respect to the separation "s" and the pitch angle of the side walls "α" are given in Table 2.2. First of all, it can be deduced from the table that the coupling capacitance is inversely proportional to the separation, and the self-capacitance decreases as the coupling capacitance increases. This effect may be attributed to the fact that the fringing fields on the side wall are directed more towards the adjacent interconnect instead of the ground plane as the separation decreases. Consequently, some fringing-field capacitance in the self-capacitance shifts to the coupling capacitance. However, as compared with the differences in capacitances shown in Table 2.2, the coupling capacitance increases more than the reduction in the self-capacitance.

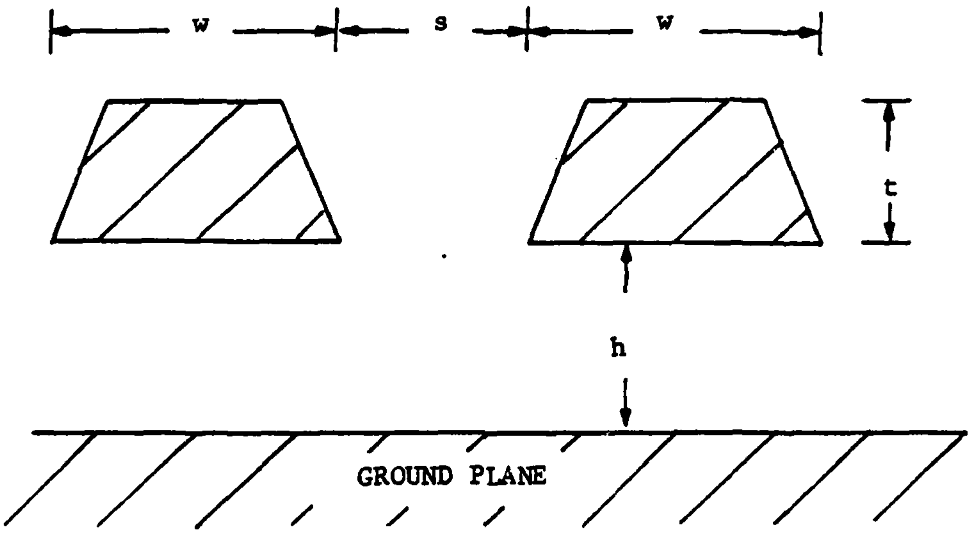


Figure 2.8 Two-dimensional, cross-sectional configuration of two interconnects.

TABLE 2.2 Self- and Coupling Capacitance vs Pitch Angle and Separation.

	* pitch angle (α)				
s/h	60°	70°	75°	80°	90°
1.0	22.332 6.729 (4.266)**	22.667 7.984 (4.626)	22.839 8.801 (4.813)	23.018 9.811 (5.006)	23.407 13.236 (5.426)
2.0	23.469 4.145 (3.129)	23.844 4.830 (3.449)	24.036 5.236 (3.616)	24.234 5.706 (3.79)	24.660 6.964 (4.173)
5.0	25.111 1.666 (1.487)	25.613 1.922 (1.68)	25.870 2.060 (1.782)	26.136 2.209 (1.888)	26.710 2.555 (2.123)
10.0	25.968 0.659 (0.63)	26.573 0.759 (0.72)	26.885 0.812 (0.767)	27.208 0.866 (0.816)	27.911 0.987 (0.922)

* The capacitances are in units of 8.854x10^{-14} Farad/cm, and the dielectric constant of the homogeneous medium is 4.0.

** The decrease in the self-capacitance as a result of the adjacency of another interconnect. The reference capacitance is the self-capacitance of a single interconnect.

Next the pitch angle seems to have a greater effect when the separation is small. For example, at s/h = 1, the coupling capacitance with a 90° pitch angle more than doubles that with a 60° pitch angle; whereas at s/h = 10, the difference in coupling capacitances is about 50%. This is due to the fact that the closer the two interconnects, the more strongly coupled they are, and the more evident is the change in pitch angle. Since the design rules of the separations between adjacent interconnects are usually dictated by the consideration for metal migration, it may not be necessary to consider the pitch angle effect. However, this ability to take into account the pitch of the side walls of interconnects is helpful if detailed information is needed. Subsequently, in all further capacitance computation of interconnects, a 90° pitch angle will be assumed.

2.7.3. Comparison of knock-knee and cross-over configurations

In the channel routing problem, there is some concern about whether the knock-knee or the cross-over configuration will have smaller coupling capacitance. In Fig. 2.9 and Fig. 2.10, both the knock-knee and the cross-over configurations of two interconnects in different levels are shown. For the sake of simple computation, the thickness of interconnects is assumed to be infinitesimally thin. As discussed previously, if the thickness is included, the coupling capacitance will be increased due to the increased side wall areas. The calculated capacitances for these two cases are listed in Table 2.3 to Table 2.6. For simplicity, both interconnects are assumed to have the same width and length. The parameter "l" is defined comparably in both knock-knee and cross-over cases as shown in Fig. 2.9 and Fig. 2.10. C_{11} is the self-capacitance of the lower interconnect, and C_{22} is that of the upper one. More important, the coupling capacitance between the two interconnects is C_{12}. Comparing Table 2.3 with Table 2.4, or Table 2.5 with Table 2.6, it is apparent that either the knock-knee or the cross-over configuration has very

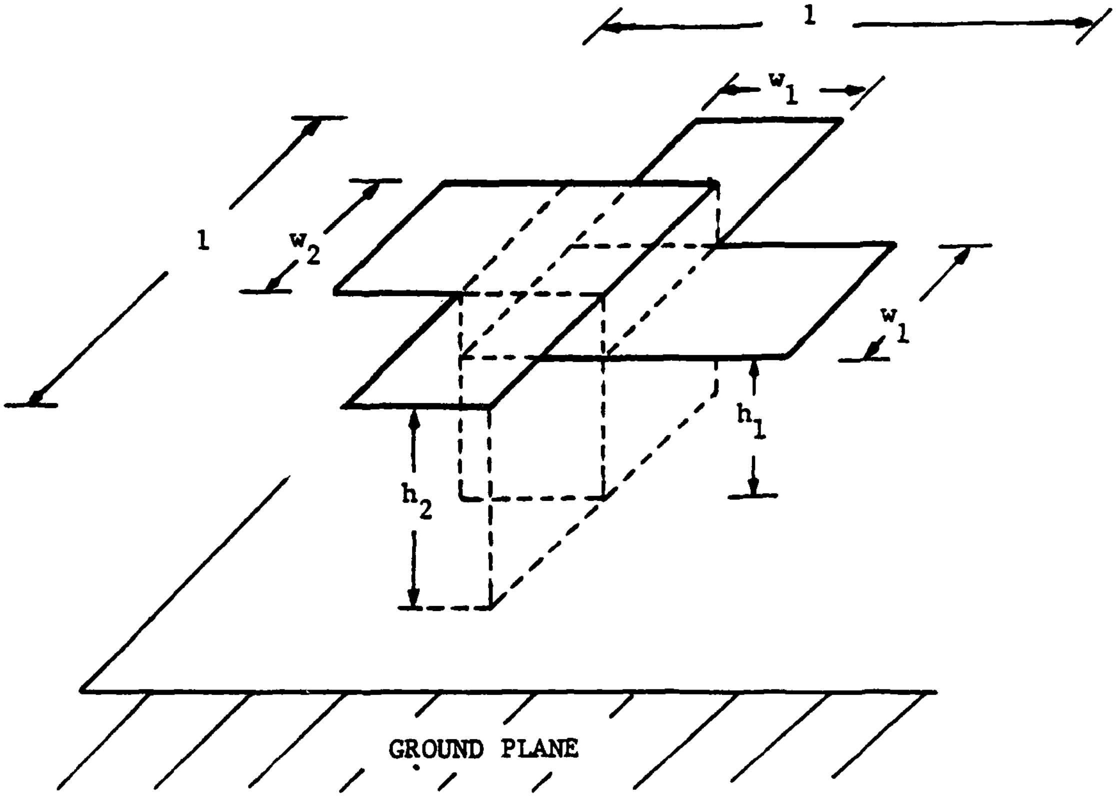


Figure 2.9 Knock-knee configuration.

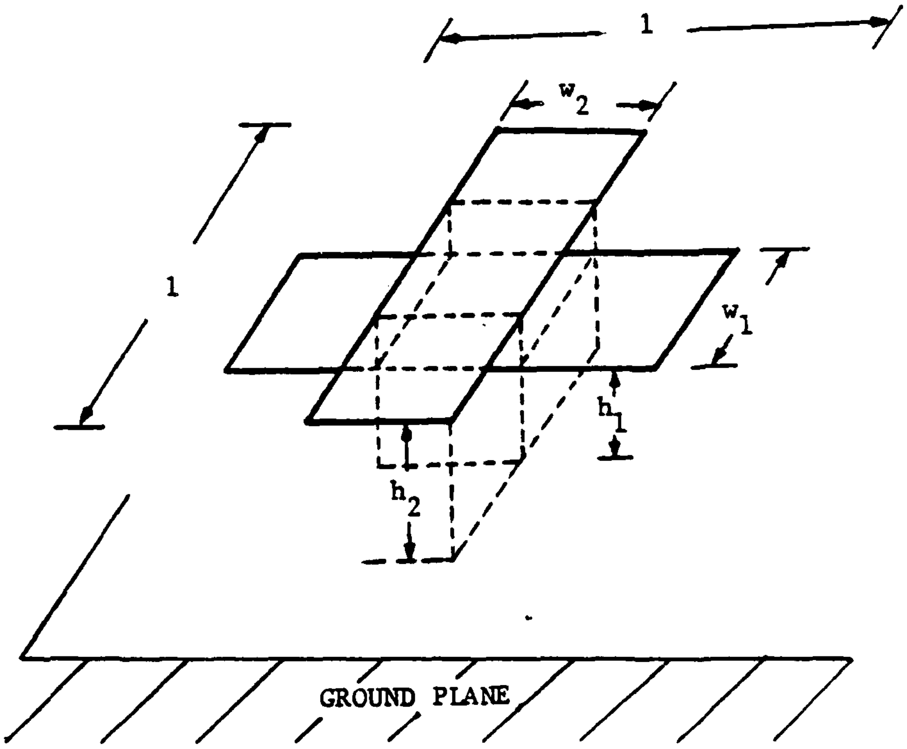


Figure 2.10 Cross-over configuration.

TABLE 2.3 Knock-Knee Capacitance.

($w_1/h_1 = w_2/h_1 = 1$, $h_2/h_1 = 2$)

l/h_1	2.0	3.0	5.0	10.0	20.0
C_{11}	6.65	9.03	14.20	27.87	56.00
C_{22}	5.59	7.32	11.08	21.22	42.47
C_{12}	2.06	2.68	3.42	4.13	4.52

TABLE 2.4 Cross-Over Capacitance.

($w_1/h_1 = w_2/h_1 = 1$, $h_2/h_1 = 2$)

l/h_1	2.0	3.0	5.0	10.0	20.0
C_{11}	6.92	9.44	14.72	28.73	57.40
C_{22}	5.78	7.60	11.44	21.87	43.56
C_{12}	2.29	3.15	4.19	5.29	5.95

TABLE 2.5 Knock-Knee Capacitance.

($w_1/h_1 = w_2/h_1 = 1$, $1/h_1 = 10.0$)

h_2/h_1-1	0.5	1.0	2.0	5.0	10.0
C_{11}	27.59	27.87	28.38	29.41	30.14
C_{22}	22.98	21.22	19.38	18.49	18.20
C_{12}	5.66	4.13	2.88	1.41	0.59

TABLE 2.6 Cross-Over Capacitance.

($w_1/h_1 = w_2/h_1 = 1$, $1/h_1 = 10.0$)

h_2/h_1-1	0.5	1.0	2.0	5.0	10.0
C_{11}	28.22	28.61	29.40	30.75	31.61
C_{22}	23.56	21.78	20.41	19.48	19.27
C_{12}	6.71	5.26	3.70	1.74	0.69

similar values of coupling capacitance between the two interconnects in different levels. Therefore, it does not matter whether the channel routing problem is solved by the knock-knee or the cross-over configuration from the viewpoint of the capacitance penalty.

In summary, the two- and three-dimensional capacitance computations can be applied to various interconnect configurations. As the dimensions of the interconnects and the proximity between them become smaller, not only the self-capacitances need more accurate computation, but also the coupling capacitances become more important. In this chapter, the numerical computation of the interconnect capacitances was discussed. Fortran programs "CAP2D" and "CAP3D" were also developed to implement the integral method discussed here. In order to easily incorporate these capacitances into an extraction program, some simpler closed-form formulas are needed instead. These will be discussed in the next chapter.

CHAPTER 3

APPROXIMATION FORMULAS FOR INTERCONNECT CAPACITANCES

3.1. Introduction

As the complexity of the present VLSI circuits increases, the interconnect parameters have become more important in both circuit design and circuit verification. In Chapter 2, numerical approaches to the computation of interconnect capacitances were discussed. However, those computations are usually quite involved and require a large amount of computation time. In extraction programs for design verification, it is impractical to go through those computations for every interconnect in the layout. Simple formulas, though not as accurate as the numerical computation, have to be derived and employed in the extraction programs. From another point of view, simple or approximate formulas can also provide some physical insight or "guidelines" for the designers. Hence, designers can take the electrical parameters of the interconnects into account in the design process and reduce the number of design iterations. In general, these approximation formulas can serve as useful tools in both design and design verification.

3.2. Self-Capacitance

In this section, the approximation formulas for the self-capacitance are discussed. The self-capacitance, by definition, is the capacitance of the interconnect with respect to the ground plane. It should be pointed out that the presence of other interconnects in the vicinity will reduce this capacitance with respect to the ground plane as discussed in Sec. 2.7, and shown in Table 2.2.

3.2.1. A review of simple formulas for self-capacitance

The simple analytic formulas for the two-dimensional self-capacitance have been discussed in the literature [20-23]. For the sake of comparision and discussion, a general review of those simple formulas is first given.

3.2.1.1. Parallel-plate formula

The parallel plate formula is a fundamental expression for the computation of capacitance. It can be written as

$$C = \varepsilon w/h \tag{3.1}$$

where ε is the dielectric constant, w is the width and h is the distance from the ground plane to the plate. Note that the formula in Eq. (3.1) is two-dimensional, and the capacitance calculated by it is in units of Farads per-unit-length. Generally speaking, this formula is applicable under the assumption that the thickness of the plate is negligibly small and the width w is considerably larger compared with the distance h. It can be considered as a first-order approximation formula and it was used successfully for the calculation of interconnect and gate capacitance before the dimensions became so small that fringing capacitance was no longer negligible.

3.2.1.2. Thin-plate formula

It is known that the charges on a conductor surface will congregate at places where there is a sharp curvature change, e.g., edges and points. Hence, the electric fields at the edges are not constant as in the middle of the plate as assumed in the parallel-plate formula. Thus, it gives rise to the "edge effect" or fringing-field capacitance of the plate. In [20] a general formula was derived from the conformal mapping method to include the edge effect of infinitesimally thin-plates. An approximate formula assuming that width w is larger than the distance h can be expressed as

$$C = \varepsilon\frac{w}{h}\left[1 + \frac{2h}{\pi w}\left(1 + \ln\frac{\pi w}{h}\right)\right], \qquad w \gg h \qquad (3.2)$$

The second term in the expression can be considered to account for the fringing-field capacitance. As w is increased larger than h, this term becomes negligible and the thin-plate formula Eq. (3.2), as expected, reduces to the parallel-plate formula of Eq. (3.1).

3.2.1.3. Chang's formula

Advancing one step forward, Chang's formula [21] furthermore takes into account the edge effect and the finite thickness of the plate. Utilizing the Schwartz-Christoffel transformation of the conformal mapping method and an ingeniously contrived transformation function, Chang was able to derive an analytic formula for two-dimensional self capacitance. It can be summarized in the following:

$$C = \varepsilon(2/\pi)\ln(2R_b/R_a), \qquad w/h \geq 1 \qquad (3.3)$$

$$\ln R_a = -1 - (\pi w/2h) - (p+1)p^{-1/2}\tanh^{-1}(p^{-1/2}) - \ln((p-1)/4p)$$

$$R_b = \begin{cases} R_b^{(1)} - [(R_b^{(1)}-1)(R_b^{(1)}-p)]^{1/2} + (p+1)\tanh^{-1}[(R_b^{(1)}-p)/(R_b^{(1)}-1)]^{1/2} \\ -2p^{1/2}\tanh^{-1}[(R_b^{(1)}-p)/p(R_b^{(1)}-1)]^{1/2} + (\pi w/2h)p^{1/2}, \quad 5 > w/h \geq 1 \\ \\ R_b^{(1)} \qquad \text{otherwise} \end{cases}$$

$$R_b^{(1)} = \eta + (p+1)/2*\ln\Delta$$

$$\eta = p^{1/2}(\pi w/2h) + (p+1)/2[1 + \ln(4/(p-1))] - 2p^{1/2}\tanh^{-1}p^{-1/2}$$

$$\Delta = \max(\eta, p)$$

$$p = 2B^2 - 1 + [(2B^2-1)^2-1]^{1/2}$$

$$B = 1 + t/h$$

The parameters w,h,ε are as defined in Eq. (3.1), and "t" is the thickness of the plate. It should be pointed out that this formula, though complicated, is rather accurate. In comparison with the two-dimensional numerical computation [4], its percentage errors are all within ±2% [21] for different combinations of w/h and t/h. Still this formula is considered too complicated to be employed in the extraction programs.

3.2.1.4. Elmasry's formula

In line with the idea to provide an easy, simple formula in the preliminary design phase and in the CAD programs, another simple formula [22] was proposed. It can be expressed as

$$C = \varepsilon\frac{w}{h}\left[\,1 + 2\frac{h}{w}\ln\left(1+\frac{t}{h}\right) + 2\frac{t}{w}\ln\left(1+\frac{w/2}{h+t}\right)\right] \qquad (3.4)$$

Note that the parameters are defined as before. Physically the second and the third terms in the above expression correspond to the capacitances resulting from the charges on the side walls and the top wall of the plate, respectively. Since this is an empirical formula, the factor 2 in front of the second and the third terms can be adjusted empirically according to the experimental data or the data from numerical computation for a better correlation with the "true" capacitance value.

All in all, the above formulas can be applied in the calculation of the self-capacitance in a homogeneous medium. Chang's formula is accurate enough but still too complicated to be incorporated in the extraction programs. Elmasry's formula is not sufficiently accurate, thus, another approximate formula, the "cylindrical approximation formula", is proposed.

3.2.2. Cylindrical approximation formula

3.2.2.1. Formulation

As a preliminary, let us consider the capacitance of an interconnect whose cross-section is a square with side t and a distance h over the ground plane. It is known that the capacitance is determined by the geometric shape, the perimeter and the distance from the ground plane. Hence, the capacitance per unit length of the square, C_s, will be a function of t and h. Since the capacitance per unit length of a conductor over a ground plane with a circular cross-section is given by the relatively simple expression [15]

$$C_c = \frac{2\pi\varepsilon}{\ln\left(\frac{d+(d^2-a^2)^{1/2}}{a}\right)} \tag{3.5}$$

where "a" is the radius of the circle, and "d" is distance from its center to the ground plane, then intuitively, we might employ Eq. (3.5) as an estimation for the capacitance C_s with a proper choice of parameters. Next we examine a few possible choices for these parameters.

Consider the inscribed circle to the square; that is, $a = t/2$, and $d = a + h$. Because like charges repel on the conductor, they will keep as far apart as possible. Consequently, smooth surfaces like the circle will have less charge accumulated near the ground plane than the square which has sharp corners for the congregation of charges. Also, the "inscribed" circle has a perimeter πt compared to a perimeter of 4t for the square. Due to this shape factor plus the smaller perimeter of the circle, it is expected that the "inscribed circle approximation" will always underestimate the capacitance per unit length of the square configuration.

Next let us consider the conductor with a rectangular cross-section, as shown in Fig. 3.1 with width w and thickness t. Also, shown in Fig. 3.1 is a cylindrical conductor whose ends are circular with a radius t/2. For this conductor the total "side-wall capacitance" can be estimated with Eq. (3.5), where $a = t/2$ and $d = t/2 + h$. In the center part of the conductor, a uniform charge distribution is assumed throughout the bottom side of the conductor, so that the capacitance of this section can be computed with the parallel-plate formula of Eq. (3.1). It was found by trial and error that the capacitance of the cylindrical conductor closely approximated the capacitance of the rectangular conductor when a section of width t/4 was taken from each end of the rectangle and replaced with a semicircular section of radius t/2, as shown in Fig. 3.1. The total perimeter of the semicircular ends is πt while the total perimeter removed from the rectanngle is 3t. Thus, there is a close

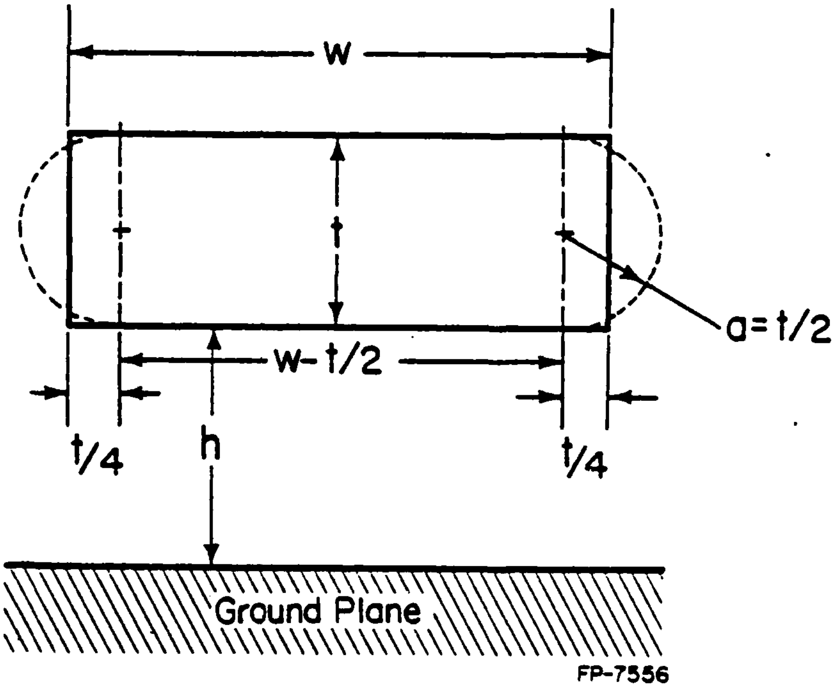


Figure 3.1 Conductor with a rectangular cross-section.

match. With the above assumptions and for the case $w \geq t/2$, the capacitance per unit length of the rectangular conductor can be estimated with the simple formula

$$C_2 = \varepsilon[\frac{w-t/2}{h} + \frac{2\pi}{\ln[1+2h/t+(2h/t(2h/t+2))^{1/2}]}], \quad w \geq t/2 \tag{3.6}$$

When $w < t/2$, it is not possible to remove sections of width $t/4$ from each end. Although this case is uncommon in integrated circuits, it was found that a reasonable approximation to the capacitance could be obtained by using the parallel-plate calculation for the entire width and modifying the circular capacitance approximation in order to account for the fringing capacitance and to maintain continuity with Eq. (3.6). The resulting formula for estimating the total capacitance per unit length in this case is

$$C_3 = \varepsilon[\frac{w}{h} + \frac{\pi(1-0.0543t/2h)}{\ln[1+2h/t+(2h/t(2h/t+2))^{1/2}]} + 1.47], \quad w < t/2 \tag{3.7}$$

In both Eq. (3.6) and Eq. (3.7), the second term attempts to account for the additional capacitance due to fringing effects at the edges of the conductor. In the case of two layers of media, the first term, parallel-plate term, and the fringing field term are weighed with different dielectric constants. This will be discussed later.

3.2.2.2. Results and comparison

Since the accuracy of Chang's formula has been established in comparison with the two-dimensional numerical computation, the capacitance values generated by it are used as the "true" capacitance in subsequent comparisons. First of all, the normalized capacitances by the "cylindrical approximation formula" are compared with those by Chang's formula in Fig. 3.2. Those capacitances are normalized with respect to the parallel-plate capacitance, and they are plotted in Fig. 3.2 for t/h = 0.1, 1, and 10. As w/h increases, all the normalized values approach constant 1. In other words, the capacitances, as expected, approach the parallel-plate capacitance when the width is relatively larger than the distance from the ground plane. For t/h = 10, the cylindrical approximation is within 5% of the actual capacitance, and for t/h = 1, the error can be as large as 10%, but only for w/h << 1. For t/h = 0.1, the error is within 20%.

Furthermore, for a more detailed comparison of various approximation formulas, the percentage errors of the capacitance values computed by the thin-plate formula, Elmasry's formula, and the cylindrical formula, are plotted in Fig. 3.3, Fig. 3.4, and Fig. 3.5 for t/h = 0.1, 1, and 10, respectively. The referenced "true" capacitance is obtained by Chang's formula Eq. (3.3). Note that the thin-plate formula and Elmasry's formula are logarithmic-type approximations, and the errors are relatively larger for smaller w/h. While the cylindrical formula is a reciprocal logarithmic approximation, the error trend is reversed for smaller w/h. Hence, the error for

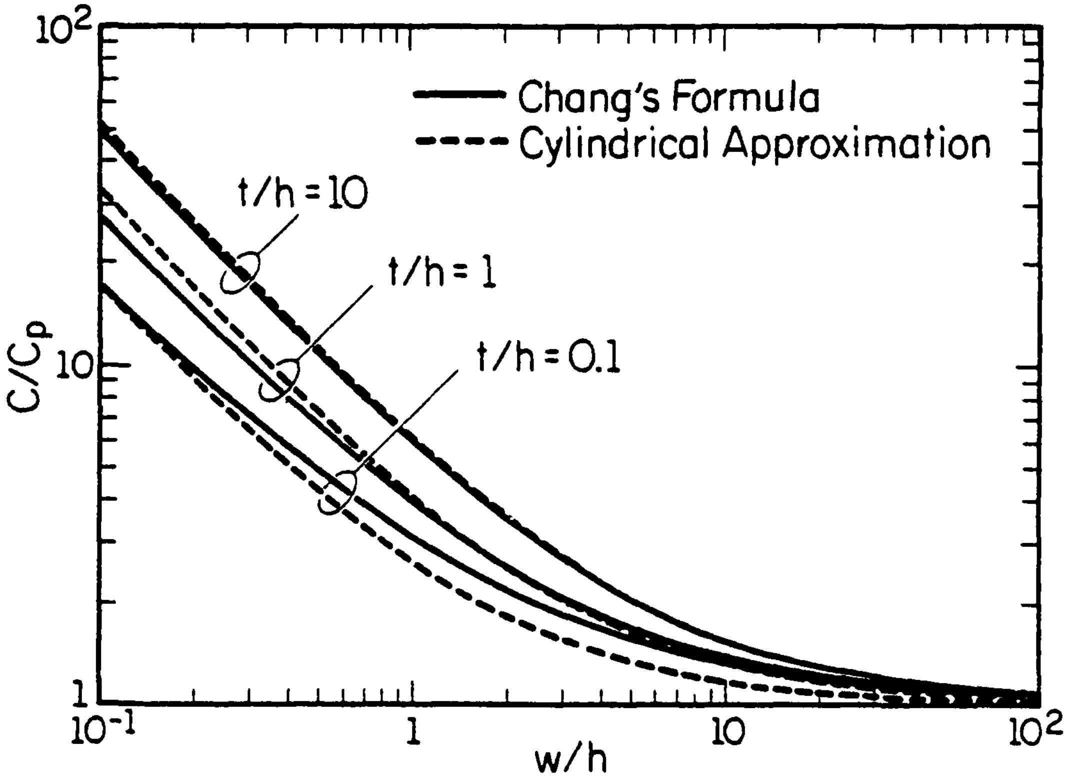


Figure 3.2 Normalized capacitance comparison.

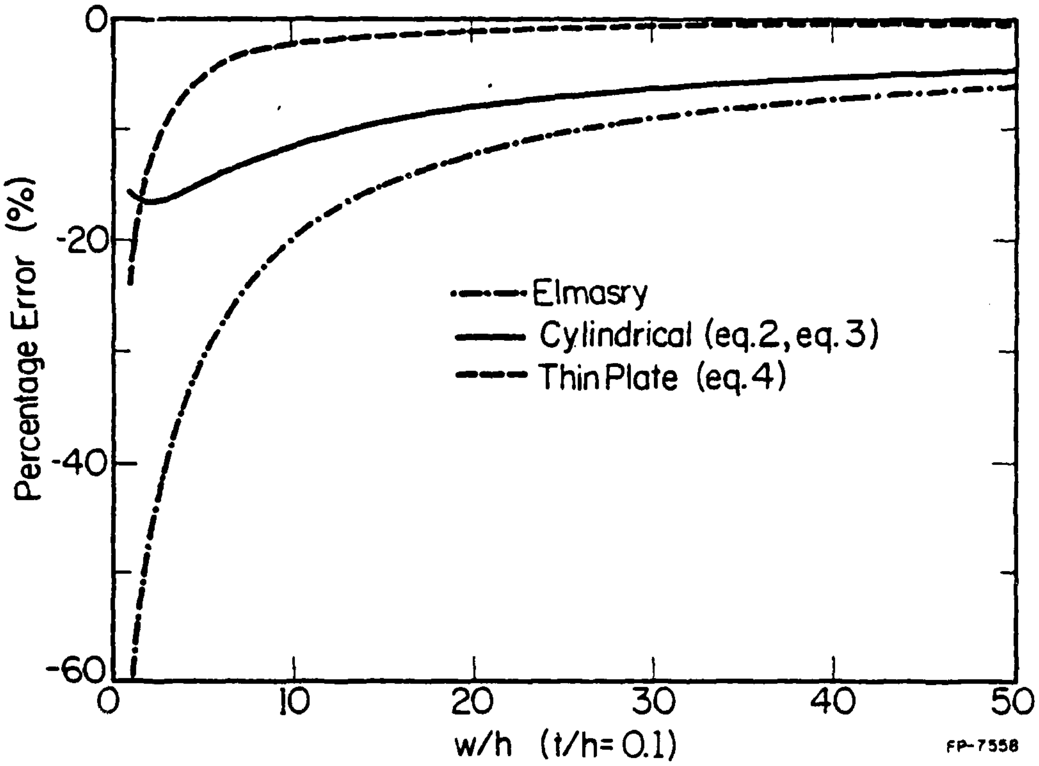


Figure 3.3 Comparison of errors of the approximate formulas (t/h = 0.1).

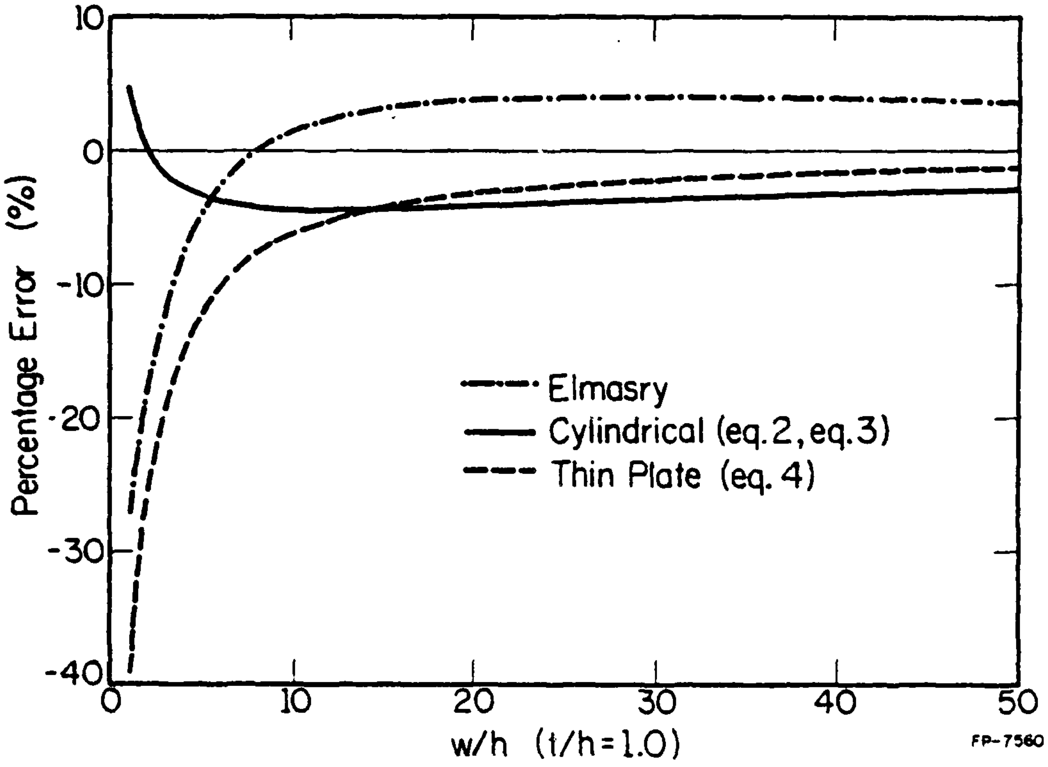


Figure 3.4 Comparison of errors of the approximate formulas (t/h = 1).

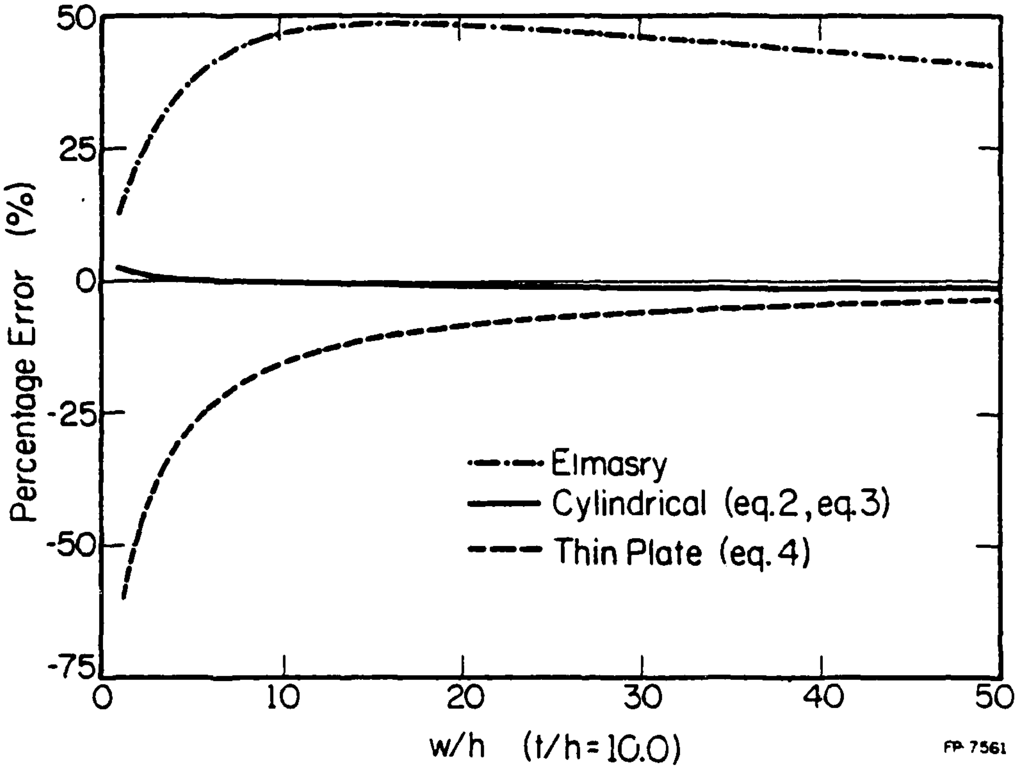


Figure 3.5 Comparision of errors of the approximate formulas (t/h = 10).

the cylindrical approximation is bounded within a smaller range and is more constant over the range. For example, at t/h = 1, the cylindrical approximation has an error within ±5% for w/h from 1 to 50. As for the other two approximate formulas, the errors are 30% and 40%. However, for the extremal case, t/h = 0.1, the thin-plate approximation, Eq. (3.2), generally has less error than the cylindrical approximation. It may be inferred that, for this case, the charges are distributed more on the bottom side of the interconnect, so that the infinitesimally thin-plate approximation is more realistic than the cylindrical approximation. Consequently, Eq. (3.2) is a better approximation is this case. For general cases, the cylindrical approximation formula compares fairly favorably with the other approximate formulas.

3.2.2.3. Two layers of media

If the interconnect is situated within a passivation layer, e.g., SiN, on top of the oxide, SiO_2, then the difference of the dielectric constants has to be taken into account in the calculation of the capacitance. This can be accomplished by selecting different weighting factors for the "parallel-plate" term and the "fringing" term in Eq. (3.6) an Eq. (3.7). Consequently, for two layers of dielectric media, the total capacitance is

$$C_t = \varepsilon_{ox}[(\text{parallel-plate term}) + \frac{\varepsilon_{avg}}{\varepsilon_{ox}}(\text{fringing term})] \qquad (3.8)$$

where $\varepsilon_{avg} = (\varepsilon_{ox} + \varepsilon_{SiN})/2$

In order to examine the errors, two-dimensional numerical computation, e.g., "CAP2D", is performed to calculate the actual capacitance. In this calculation, it was assumed that the passivation layer is thick enough so that the field lines passing through the outer boundary are negligible. Thus, the thickness of the passivation layer was considered to be infinite in the calculation of the actual capacitance. The percentage error between this capacitance and the capacitance obtained from the simplified formulas is shown in Table 3.1. The cylindrical approximation means the formulas of Eq. (3.6) and Eq. (3.7) are modified according to Eq. (3.8), and the thin-plate approximation means Eq. (3.2) is modified according to

TABLE 3.1 Percentage Error in Capacitance Estimation with Two Dielectric Layers.

	Percent Error			
	cylindrical approximation			thin plate
w/h	t/h = 10	t/h = 1	t/h = 0.1	t/h = 0.1
1.0	−5.26	5.82	−18.24	−28.12
2.0	−5.74	0.50	−20.73	−16.53
10.0	7.06	−6.40	−16.30	−4.63

* The dieletric constants are : $\varepsilon_{ox} = 3.9$, $\varepsilon_{SiN} = 7.5$.

Eq. (3.8). Note that the errors of the cylindrical approximation at $t/h = 0.1$ are relatively large compared with those of the thin-plate approximation, except at $w/h = 1$. As discussed previously, modified Eq. (3.2) is a better approximation than the modified formulas of Eq. (3.6) and Eq. (3.7) in this case. However, for the other cases, $t/h = 1$ and 10, the percentage errors of the cylindrical approximation are well within ±10%. Despite the already-existing errors in Eq. (3.6) and Eq. (3.7), the simple weighting factors in Eq. (3.8) approximate the case of two dielectric layers satisfactorily. Therefore, the simple formulas, except Eq. (3.3), may also be applicable for the capacitance of interconnect in VLSI circuits when the passivation layer is taken into account.

3.2.2.4. Cylindrical approximation on the overlap capacitance

The overlap capacitance between the gate and the source or the gate and the drain in an MOS device is an important parameter in circuit simulation. It affects the MOS model in the circuit simulation program and impacts on the circuit response. Typically it was estimated by the parallel-plate formula as Eq. (3.1). As a side topic, it was found out that the cylindrical approximation formula can also be applied to the estimation of the overlap capacitance.

A simple model and formula were derived for the overlap capacitance in an MOS device in [25]. Assuming the side walls of the gate and those of the source are vertical boundaries, as shown in Fig. 3.6, the simple formula derived in [25] was expressed as

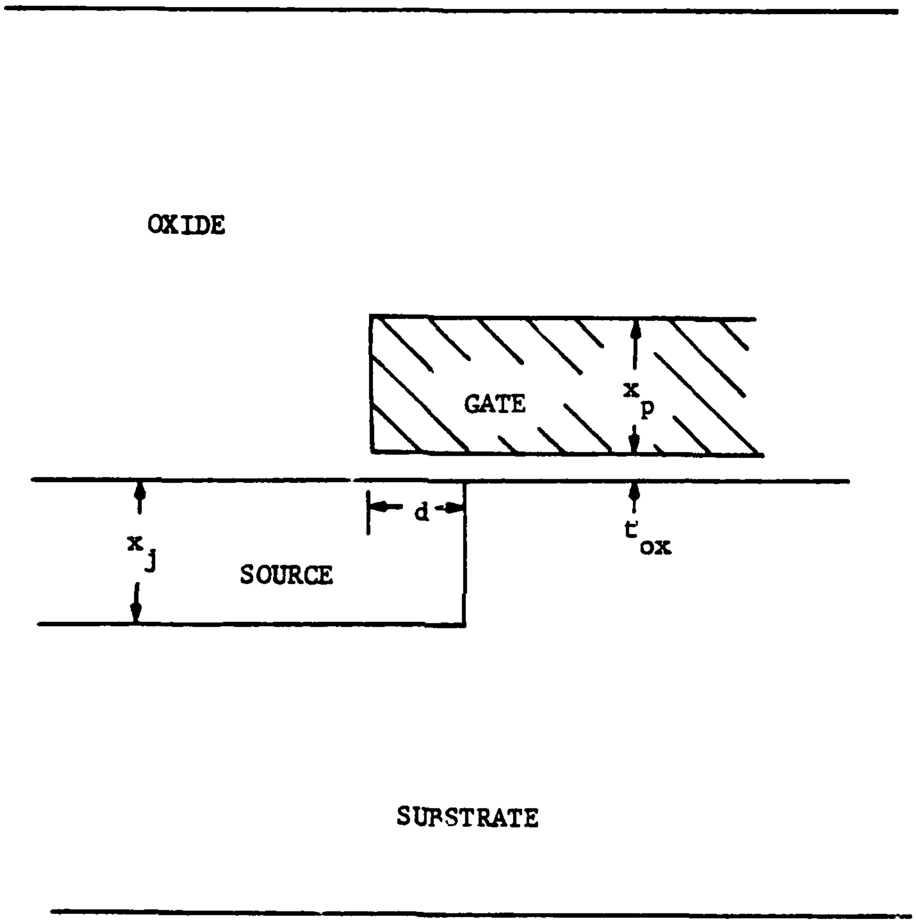


Figure 3.6 Overlap capacitance model.

$$C_{ov} = \frac{2}{\pi}\varepsilon_{ox}\ln(1+\frac{x_p}{t_{ox}}) + \varepsilon_{ox}\frac{(d+\Delta)}{t_{ox}} + \frac{2}{\pi}\varepsilon_{Si}\ \ln(1+\frac{x_j}{t_{ox}}\sin\beta) \qquad (3.9)$$

$$\Delta = \frac{t_{ox}}{2}\cdot\frac{1-\cos\beta}{\sin\beta}$$

$$\beta = \frac{\pi}{2}\cdot\frac{\varepsilon_{ox}}{\varepsilon_{Si}}$$

Note that "t_{ox}" is the gate oxide thickness, "d" is the overlap width between the gate and the source, "x_p" and "x_j" are the thicknesses of the gate and the source, respectively. The accuracy of this formula is manifested in [25] by comparison with experimental data and two-dimensional device model simulation.

Employing the basic idea of the cylindrical approximation for the fringing fields of the finite thickness of the gate and the source of an MOS device, we can obtain the cylindrical approximation on the overlap capacitance as

$$C'_{ov} = \varepsilon_{ox}\frac{d}{t_{ox}} + \varepsilon_{ox}\frac{0.5\pi}{\ln(1+\beta+(\beta(\beta+2))^{1/2})} + \varepsilon_{avg}\frac{0.5\pi}{\ln(1+\gamma+(\gamma(\gamma+2))^{1/2})}$$

where $\beta = 4t_{ox}/x_p$; $\gamma = 4t_{ox}/x_j$ (3.10)

$$\varepsilon_{avg} = (\varepsilon_{Si} + \varepsilon_{ox})/2$$

The first term in Eq. (3.10) accounts for the parallel-plate component of the capacitance, the second one results from the fringing field of the gate, and the third term is due to the fringing field of the source. Because the source is embedded inside the silicon substrate, the difference in the dielectric constant between silicon oxide and the substrate is taken into account by the factor ε_{avg}.

A few selected data points comparing Eq. (3.10) with Eq. (3.9) are given in Table 3.2. The errors are mostly bounded within 1% for

TABLE 3.2 Comparison of Simple Formulas for Overlap Capacitance.

d	t_{ox}	x_p	x_j	C_{ov}^*	$C_{ov}'^*$
0.2	0.035	0.4	0.4	4.104	4.074
0.4	0.035	0.4	0.4	6.128	6.098
0.6	0.035	0.4	0.4	8.152	8.122
0.8	0.035	0.4	0.4	10.176	10.145
1.0	0.035	0.4	0.4	12.200	12.169
0.5	0.5	1.0	1.0	1.295	1.301
0.5	0.5	0.5	0.5	1.009	1.082
1.0	0.02	0.2	0.5	20.234	20.342
0.5	0.02	0.5	0.2	11.025	11.134
10.0	1.0	10.0	10.0	5.519	5.467

* C_{ov} is the capacitance calculated by Eq. (3.9), and C_{ov}' is that by Eq. (3.10).

various combinations of paramters. Thus, it can be said that the cylindrical approximation on the overlap capacitance is as good a simple formula as that proposed in [25]. Hence, the cylindrical approximation can be more generally applied to the estimation of the fringing capacitance due to the thickness of a plate.

To sum up, the "cylindrical approximation formula" seems capable of being generally applied to the estimation of the fringing field capacitance, be it due to the side walls of an interconnect, or those of a gate or a source. Besides, the errors from it are bounded within a reasonable limit, and the expressions are simple enough for the application in the extraction programs for design verification.

3.3. Coupling Capacitance

In VLSI circuits, as the design rules become more stringent interconnects are closer together; inevitably, the coupling capacitances between the interconnects become important and have to be reckoned with in the circuit simulation. By numerical computation methods, they can be evaluated rather accurately along with the self-capacitances. However, as mentioned before, it is essential to have a simple formula for coupling capacitances in order to incorporate them into the extraction programs for design verification.

Since the coupling capacitances are strongly related to the self-capacitances, it is not easy to treat them separately and derive a simple formula solely for the coupling capacitance. For example, the simple formulas derived in [24] are for the estimation of the sum of the coupling and the self-capacitances for two interconnects. However, one method for generating a simple formula for the coupling capacitance uses a "least square fit" on the data obtained from numerical computation. In [24] the simple formulas have been obtained by this method. Employing the programs described in Chap.2, i.e., "CAP2D" and "CAP3D", we can collect a few sets of data on different configurations of interconnects. Then utilizing a subroutine, we can fit different formulas with multiple coefficients to the data in search of better simple formulas.

3.3.1. Coupling capacitance between two parallel interconnects

Let us consider the configuration of two parallel interconnects in a homogeneous medium similar to the figure shown in Fig. 2.8. In the most general case, there are seven parameters to consider : the width "w", the thickness "t", and the distance "h" from the ground plane for each interconnect, plus the separation "s". In order to reduce the variables in the simple formula, let us assume that both interconnects possess the same parameters, and one of the parameters is utilized as the normalizing factor; then, in essence there are three variables. In [24] a simple formula for two interconnects with identical parameters was derived. It estimated the sum of the self- and the coupling capacitances for one interconnect. It is

$$C = \varepsilon_{ox}\{1.15(w/h) + 2.80(t/h)^{0.222} + (s/h)^{-1.34} \cdot [0.03(w/h) + 0.83(t/h) - 0.07(t/h)^{0.222}]\} \qquad (3.11)$$

The accuracy of the formula has been shown in [24] to be within ±10% of the two-dimensional numerical computation. But the coupling and the self-capacitances are not separated.

In order to obtain a simple formula solely for the coupling capacitance, the same "least square fit" method is applied on the data calculated by "CAP2D", and a similar formula is obtained :

$$C_{12} = \varepsilon_{ox}\{0.0046(w/t)^{1.67} + 0.0271(h/t)^{1.56} + (s/t)^{-1.057} \cdot [0.496(w/t)^{0.248} + 0.914(h/t)^{0.203}]\} \qquad (3.12)$$

The errors of this approximate formula, Eq. (3.12), are given in Table 3.3. Hence, Eq. (3.12) can be used in conjunction with Eq. (3.11) to evaluate the self-capacitance and the coupling capacitance individually.

3.3.2. Three-dimensional cross-over geometry

In this section the configuration of perpendicular cross-over of two interconnects in Fig. 3.7 is considered. As in the previous section, to reduce the variables considered, both interconnects are assumed to have the same width and thickness. Since the coupling between the two over the cross-over region is of interest, the

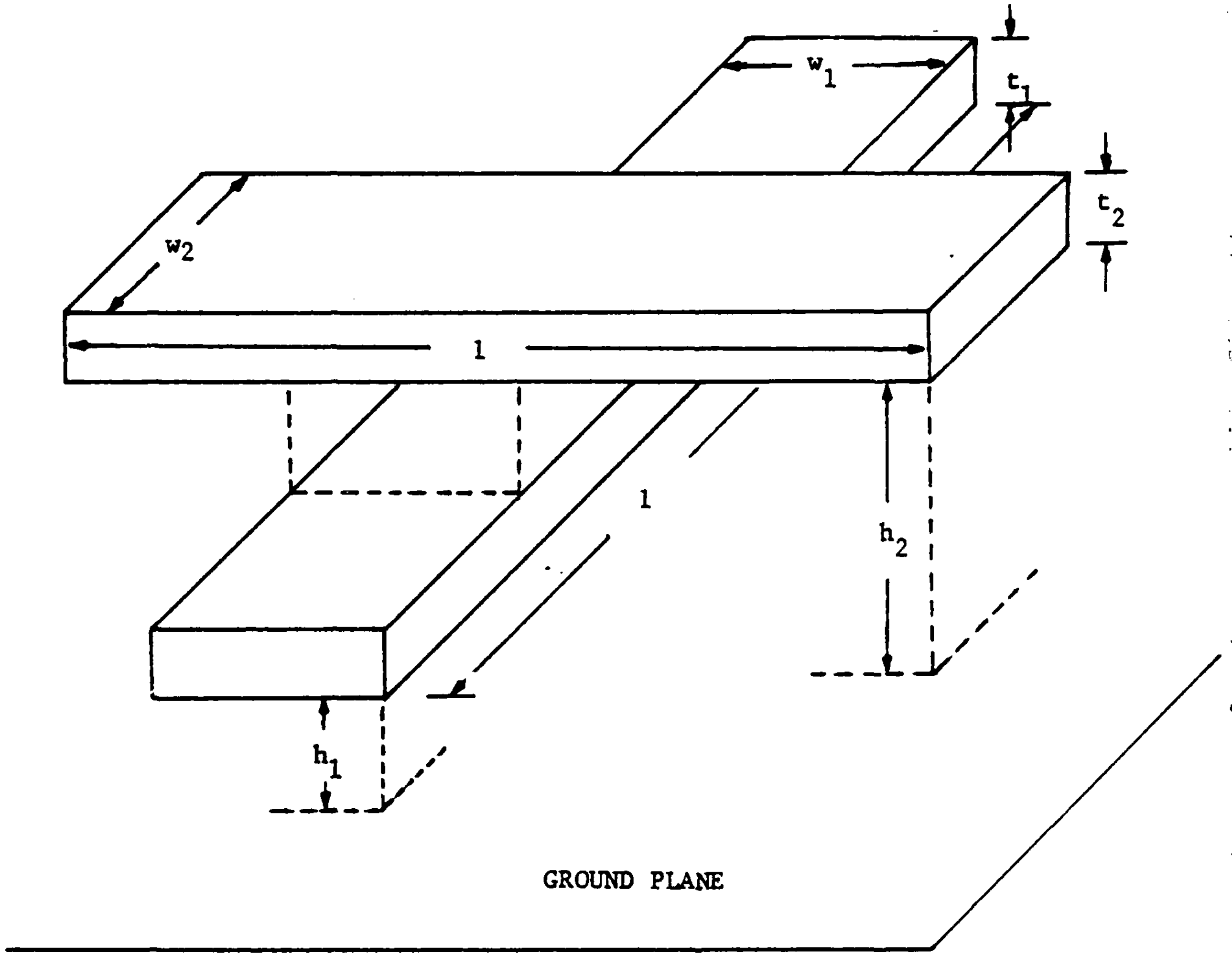


Figure 3.7 Configuration of cross-over geometry of two interconnects.

TABLE 3.3 Percentage Error of Eq. (3.12).

		s/t			
w/t	h/t	.500	1.000	2.000	5.000
1.00	.50	-3.7245	-9.5482	-3.5060	42.2329
1.00	1.00	-2.2482	-12.6091	-12.9382	20.2085
1.00	2.00	1.6377	-12.7294	-18.5078	2.8354
1.00	5.00	15.0499	-.5205	-6.5252	14.9383
2.00	.50	-1.0380	-9.9699	-8.9054	21.9106
2.00	1.00	.2764	-12.7097	-16.4034	6.5781
2.00	2.00	3.2419	-13.3229	-21.3423	-6.1157
2.00	5.00	14.4955	-3.5693	-11.8742	4.3947
5.00	.50	4.4536	-8.4302	-12.6018	5.5028
5.00	1.00	5.6243	-10.7517	-18.1644	-4.0452
5.00	2.00	8.1620	-11.3784	-22.0076	-12.5019
5.00	5.00	17.2850	-3.9135	-14.8168	-3.9239
10.00	.50	11.4921	-2.1553	-6.1729	14.9096
10.00	1.00	12.1799	-4.7770	-11.6876	5.6213
10.00	2.00	14.3737	-5.7158	-15.7960	-3.4468
10.00	5.00	22.6563	.6820	-10.2202	1.2573

lengths of the interconnects are assumed to be five times the widths to reduce further the number of variables. Using "t" as the normalizing factor, we have a total of three variables, w/t, h/t, and s/t. In this case, the parameter "h" designates the distance of the lower interconnect from the ground plane. The program "CAP3D" was employed to evaluate the coupling capacitances in this configuration, and the data are shown in Table 3.4. A few approximate formulas composed of the variables w/t, h/t, and s/t were tested. The following formula seems to have the smallest error:

TABLE 3.4 Coupling Capacitance of the Cross-Over Geometry.

		s/t			
w/t	h/t	0.500	1.000	2.000	5.000
1.000	.500	39.323	27.732	16.808	6.132
1.000	1.000	41.811	29.811	18.386	6.986
1.000	2.000	44.749	32.364	20.450	8.243
1.000	5.000	48.032	35.659	23.406	10.385
2.000	.500	78.091	60.677	41.846	19.473
2.000	1.000	84.008	65.220	45.190	21.395
2.000	2.000	89.104	70.923	50.281	24.652
2.000	5.000	98.349	79.420	57.593	29.873
5.000	.500	204.420	168.070	126.430	74.454
5.000	1.000	209.290	174.900	133.640	79.888
5.000	2.000	207.090	179.510	142.740	88.719
5.000	5.000	224.790	199.590	163.530	105.730
10.000	.500	500.330	412.840	313.010	192.930
10.000	1.000	480.290	407.360	318.000	201.730
10.000	2.000	436.340	386.570	317.970	214.240
10.000	5.000	419.600	387.050	336.120	243.620

$$C_{12} = \varepsilon_{ox}\{0.0034(w/t)^{3.248} - 5.624(h/t)^{0.0175} + (s/t)^{-0.354} \cdot [6.74(w/t)^{1.118} + 6.25(h/t)^{0.171}]\} \tag{3.13}$$

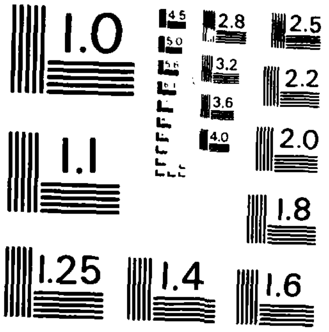
Note that the dielectric constant ε_{ox} in Eq. (3.13) is in units of Farads. The percentage errors of the approximate formula (3.13) compared with the coupling capacitances in Table 3.4 are listed in Table 3.5.

TABLE 3.5 The Percentage Errors of the Simple Formula (3.13).

		s/t			
w/t	h/t	.500	1.000	2.000	5.000
1.00	.50	3.3618	-2.7977	-3.3277	-8.9060
1.00	1.00	5.0931	-1.1252	-1.2234	-1.3110
1.00	2.00	6.5507	-.0683	-.5180	1.8643
1.00	5.00	11.1813	3.0278	1.2737	3.2677
2.00	.50	3.8913	-3.3425	-1.8682	20.9546
2.00	1.00	.4967	-6.2124	-4.8980	16.2017
2.00	2.00	-1.0536	-9.7125	-10.1856	6.9443
2.00	5.00	-4.5365	-13.8360	-15.7542	-3.9548
5.00	.50	6.2601	-1.4563	-.9216	14.2679
5.00	1.00	5.3622	-3.8641	-4.8359	8.1325
5.00	2.00	8.2874	-4.7362	-9.3735	-.9373
5.00	5.00	2.3064	-12.1174	-18.8404	-14.6741
10.00	.50	-3.4869	-8.4098	-5.3784	11.2182
10.00	1.00	1.2263	-6.5592	-6.2619	7.0149
10.00	2.00	12.2793	-.7924	-5.5665	1.4674
10.00	5.00	18.1225	.2205	-9.6661	-9.8139

Note that Eq. (3.13) has a similar structure to that of Eq. (3.11). Also the maximum percentage errors are +20% and -18%, and the nomial errors are around 10% as seen in Table 3.5. Since more levels of interconnection are needed in increasingly complicated VLSI circuits, Eq. (3.12) would be helpful to estimate those coupling capacitances between levels of interconnects.

AD-A142 381   MODELING AND EXTRACTION OF INTERCONNECT PARAMETERS IN   2/3
VERY-LARGE-SCALE INTEGRATED CIRCUITS(U) ILLINOIS UNIV
AT URBANA COORDINATED SCIENCE LAB C P YUAN AUG 83
UNCLASSIFIED R-994 N00014-79-C-0424   F/G 9/5   NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

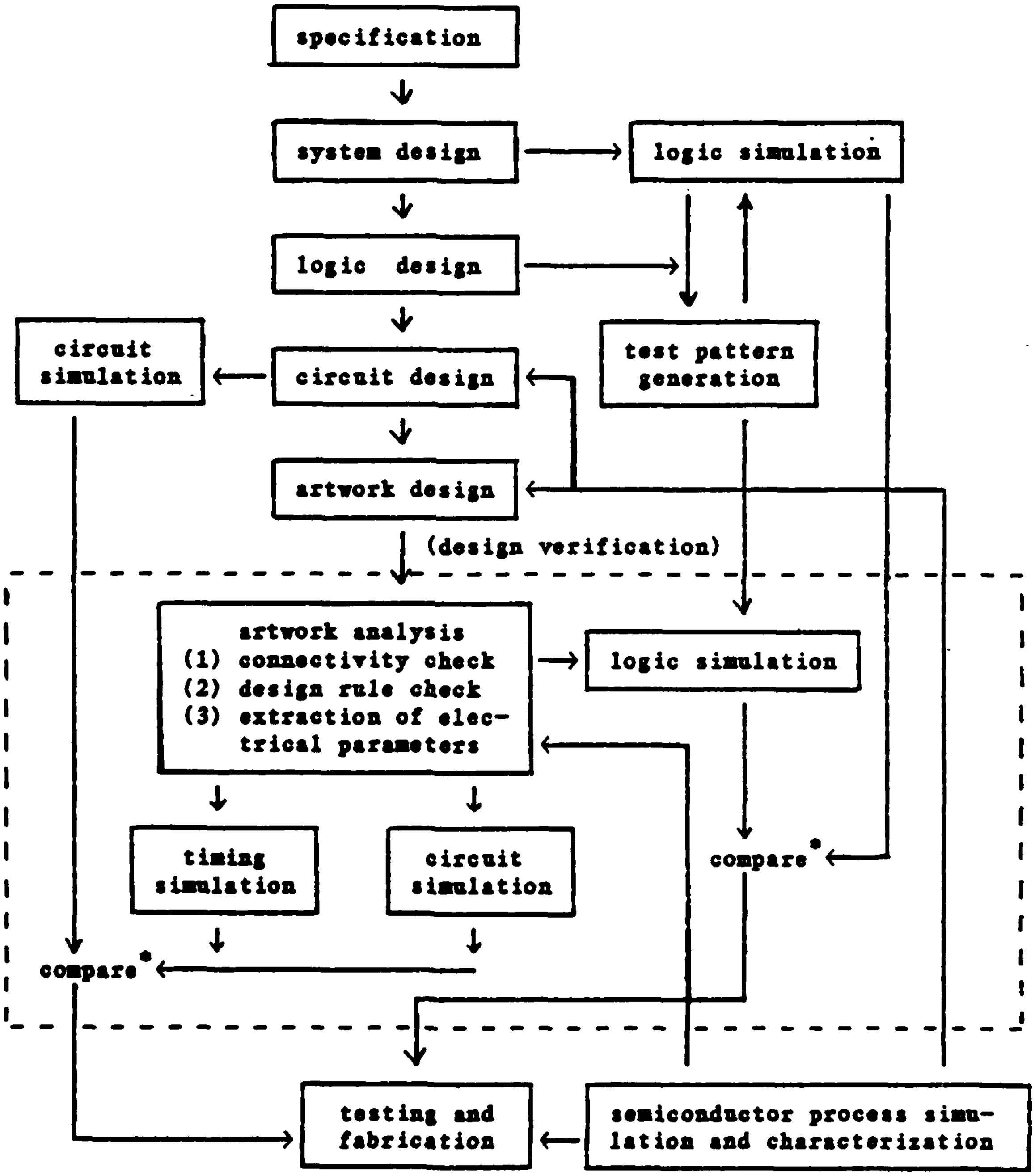
The approximation formulas discussed in this chapter can be viewed as a means to incorporate the capacitance information computed in Chapter 2 into the extraction programs. Another method, e.g., table look-up, can also be used instead of using the "least square fit" on the data to find an approximation formula. However, these approximation formulas may also provide a physical interpretation and are simpler to handle. Thus, they are also useful in the circuit design.

CHAPTER 4

NETWORK EXTRACTOR

4.1. Introduction

The design process for LSI or VLSI circuits still is not fully automated today, but contains more or less manual design steps. These manual and the interactive layout techniques which are used to optimize the layout design inevitably introduce human errors and inaccuracy into the design. Consequently, it is imperative to fully check the layout and verify the design. With the ever-growing circuit complexity, some automated tools for design verification are necessary. Figure 4.1 shows a process flow of design and the interaction of the design tools at various stages. The first step in the design verification is the artwork (or topological or mask or layout) analysis. This step basically transforms the layout information back to the circuit information for rechecking the circuit performance. It can generally be classified into three distinct, but highly interrelated categories [33] : (i) connectivity checks, (ii) design rule checks, and (iii) the calculation of electrical parameters. The connectivity checks examine the physical connection of the layout elements to see if there are short circuits or open circuits. Design rule checks basically examine the tolerances between geometric elements to see whether they are in accordance with the process



* If the results are not comparable, iterations are needed.

Figure 4.1 Computer-aided design.

requirements. The calculation of electrical parameters should include both the transistors and the interconnect parameters. For example, the channel width, the channel length, the areas of source and drain, and the overlap capacitances, etc., are important electrical parameters for transistors. The capacitance and the resistance are the electrical parameters for the interconnect. In the literature, most network extractors that have been developed concentrated on the connectivity checks and the design rule checks [27,36], or the efficiency of the algorithms [32,34,35], or the hierarchical combination of the computer—aided tools [32,37]. Not too much effort has been spent on the extraction of the electrical parameters. Hence, in the network extractor developed here, our main goal is to concentrate on the third aspect of the layout analysis, i.e., the calculation of electrical parameters, especially the interconnect capacitances.

In the face of the growing circuit complexity, the lengths of the interconnects on the chip have to increase as pointed out in [18]. Due to the smaller sizes of the transistors and this increase of interconnects, the electrical parameters of the interconnects become essential. Furthermore, those parameters usually are not taken into account in the design phase by the interactive layout technique, so it is important to include them in the design verification to ensure satisfactory circuit performance. In the previous chapter, approximate formulas for the interconnect capacitances have been discussed. They are used in the extractor discussed in this chapter to calculate the interconnect capacitances.

4.2. General Properties of the Extractor

The extractor* is coded in PASCAL, and some important properties of it can be summarized as follows :

(1) CIF format : The input of the layout information to the extractor should be in CIF (Caltech Intermediate Form) format as defined in [38]. However, not all the definitions of CIF are recognized, e.g., round flash and some user-defined extensions are not known to the extractor and errors will result. Nevertheless, there is an option "-u" in running the package to ignore the errors resulting from user-defined extensions. It should also be pointed out that the coordinates of the input rectangles should be integers; in other words, the fraction numbers after the decimal points in the coordinates will be chopped off.

(2) NMOS technology : The extractor assumes NMOS technology and the design rules defined in [38].

(3) Manhattan geometry : Since most layouts do not contain angular geometries and for the sake of simplicity in the extraction process, all the geometries on the layout should be rectilinear with respect to either the x or y axis. However, the original version of the first part of the code from Mike Graf was intended to include also the non-Manhattan geometry. Conse-

*The first part of the code which reads in CIF file and sets up the 4-d binary trees for the rectangles on every mask is due to Mike Graf [40].

quently, there is some redundancy in the definition of the rectangle record.

(4) <u>Rectangle-based data structure</u> : The basic entity of the data structure is a rectangle on a mask. All the rectangles on the same mask are then linked together to form a 4-d binary search tree with right thread. [26,40]

(5) <u>Transistor configuration</u> : It is assumed that the overlap of diffusion and polysilicon regions constitutes a transistor channel. For the sake of simplicity the extractor only recognizes basically two configurations of transistors, i.e., those formed by one diffusion rectangle and one polysilicon rectangle or those by two diffusion rectangles and one polysilicon rectangle. Specialized transistor configurations, e.g., those long, serpentine channel regions of output buffers, are not recognizable. Warning messages will be generated for these unrecognizable overlapped regions of diffusion and polysilicon in the output log file. Functionally, the extractor will categorize all the transistors into three different types of transistors, namely, load, drive and pass transistors. The channel lengths and the widths of the transistors are evaluated and included in the output SPICE file.

(6) <u>Output format</u> : The transistors, resistors and capacitors are reported in a form compatible with the SPICE input format. Also the node numbering is automatically done in the extractor. For example, the node "0" is reserved for the ground node, and "5"

is for the power node. Every center node of the T equivalent lumped circuit for an interconnect has a node number 5xx in order to distinguish it from other node numbers which start from "6" and are incremented one by one.

(7) <u>Parameter file</u> : All the process parameters are stored in the "constant.h" file. For example, the field oxide thickness, the thickness of the interconnects, sheet resistance of polysilicon, metal and diffusion, etc., all should be given in this file, and the extractor will fetch the information from this file in the process of extraction. In order to screen off small resistances and capacitances from short interconnects, two parameters "CLIMIT" and "RLIMIT" can be set in this file such that the interconnect resistances and capacitances smaller than those values will not be reported. Other parameters in "constant.h" include "WIREGAP" and "BRANCHGAP", which are used in the computation of coupling capacitances, and will be discussed in a later section.

4.3. Data Structure

In this section, the data structure used in the program is discussed. Since the program is coded in PASCAL, the "record" definitions of "rect", "transistor", "cntlist", "wire", etc., are instructive to illustrate the data structure.

The record of a rectangle in the program is defined as :

```
rect*    = record
```

```
            key   : array[0..3] of real;  (* four keys :
                                             key[0] = xmin
                                             key[1] = ymin
                                             key[2] = -xmax
                                             key[3] = -ymax  *)
            mask  : masklevel;    (* -7...7*)
            lson,         (* left son in tree structure *)
            rson  : rectptr;      (* right son in tree structure *)
            ndisc : integer;      (* discriminator of successor----
                                     only valid if rson is a thread *)
            lapdicator : integer; (* an indicator *)
            end;
  rectptr =^rect;
```

The parameter "mask" in the above definition indicates the mask level on which this rectangle resides. The mask levels are designated as follows :

1 = diffusion
2 = ion implant
3 = polysilicon
4 = contact cut
5 = metal
6 = buried contact
7 = overglaseing

Usually we are concerned with only four mask levels, i.e., diffusion, polysilicon, contact cut and metal. The levels of buried contact cut and overglassing are neglected in this extractor. Note also that, as a rule, the pointer to a certain data type is named by appending "ptr" to the name of that data type. Thus, "rectptr" is the pointer to rect, and "rectptr =^rect" is PASCAL's way of specification.

*Only those specifications which are utilized in this extractor are listed here. Others concerning non-Manhattan geometries and the optimization of the binary trees are not included, though they are in the program.

A 4-d binary search tree with right thread is built for the rectangles on each mask level. Those three parameters "lson", "rson" and "ndisc" are used to link the rectangles in a binary search tree. Each rect record is stored as a node in the tree, and lson, rson are its two pointers which are either null or point to another node in the tree. (Or each pointer can be considered as specifying a subtree to this node.) The discriminator of a node is an integer between 0 and 3 in a 4-d binary tree. Its purpose is to specify the relative orientation of the node to its sons and its parents. It is determined as follows : The root of the tree has discriminator 0. Its two sons have discriminator 1, and so on. The discriminator of level 4 becomes 0 again. The cycle repeats between 0 and 3. Generally speaking, for any node P in the 4-d binary tree, let j be its discriminator, then for any node Q in the left branch of P, it is true that $key_j(Q) < key_j(P)$; likewise, for any node R in the right branch of P, $key_j(R) > key_j(P)$. In case $key_j(Q) = key_j(P)$, then the next key, (j+1)mod4, is compared, until the order is decided. If all keys are equal, then the two rects are identical, no node will be inserted into the tree. In aiding the search process, a right thread is added to the tree [39]. A negative mask value in the rect record indicates that the right son of this node is a right thread not a true son, and "ndisc" contains the discriminator of this successor which is linked by this thread. It should be noted that the parameter "ndisc" is only valid when the right son of the node is a right thread, i.e., the mask is a negative integer.

In Fig. 4.2 an example of some rectangles on a mask and its corresponding 4-d binary search trees are depicted. The dotted lines in the tree represent the right threads and the solid lines are "true" links of the trees. Those right threads should always point to previous nodes, i.e., nodes that have been visited before the current node in the search process. A "nil" at the rightmost end of the tree signals the end of the search. Note that the binary search trees are different depending on the insertion order of the rectangles. To be more specific, two different insertion orders and their resulting binary search trees are shown in Fig. 4.2. These two 4-d binary search trees all represent the rectangles shown in Fig. 4.2, yet they have different structures. Since the efficiency of the search process obviously depends on the structure of the binary tree, the insertion order which is random in this extractor affects the efficiency of the search process. One attempt to optimize the search process is to try to balance the heights of those binary trees. This was included in the original version by Mike Graf, but is not considered here. Finally, the parameter "lapdicator" is an indicator for the subsequent search process to indicate that this rectangle has been used already.

The transistor records are defined as :

```
transistor = record
                ttype   : nmostype;     (* three types :
                                           load, drive, pass *)
                width,
                length  : real;
                drain,
                source  : remrectptr;   (* source and drain rects *)
```

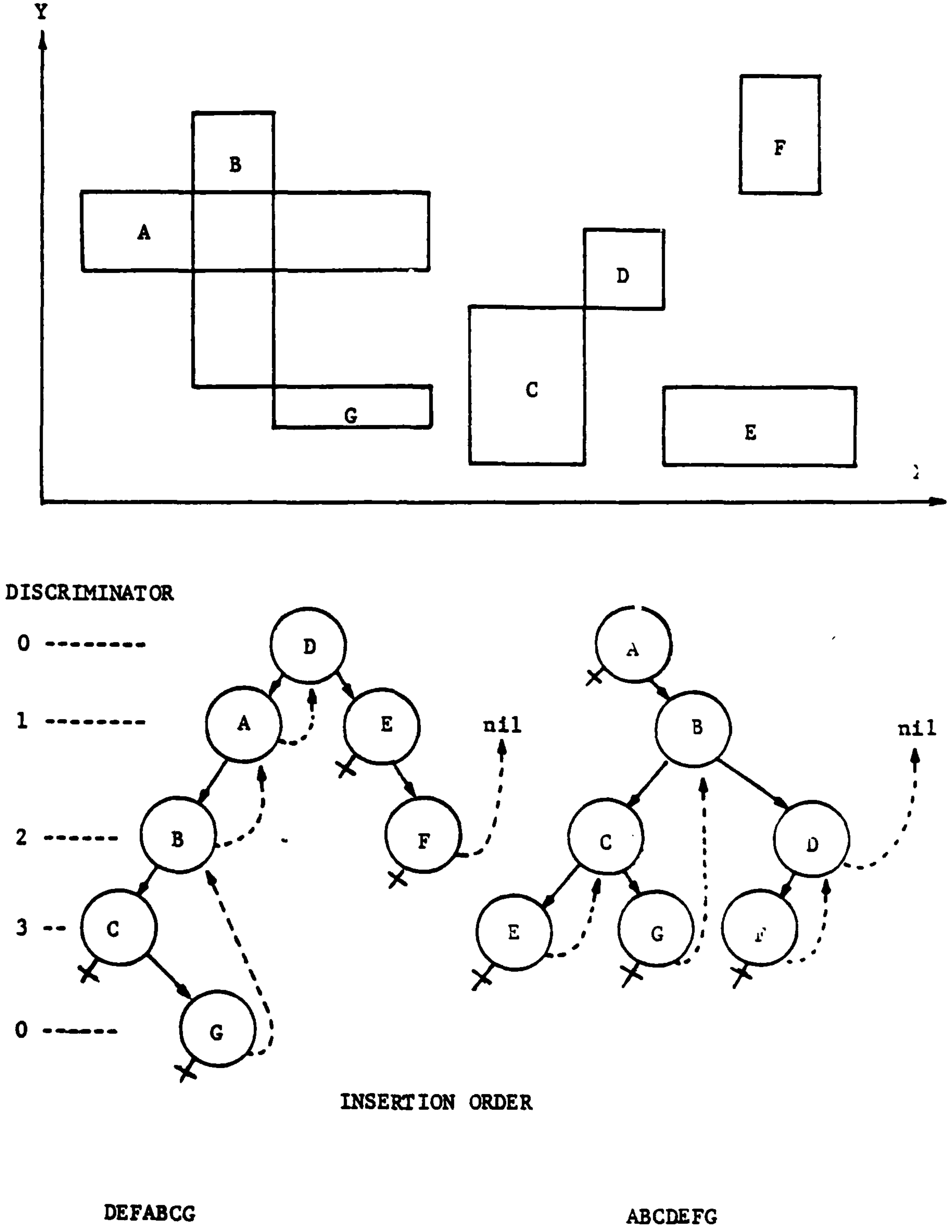


Figure 4.2 An example of 4-d binary search trees.

```
                gate    : pdlaptr;
                dnode,
                snode,
                gnode   : integer;      (* node numbers *)
              end;
tranptr =^transistor;
```

Most of the parameters defined in the above record are self-explanatory. Those diffusion rectangles residual of the overlap regions of the transistor channels are designated as "remrects". All of these remnant rectangles are built into a remnant tree as before. "Gate" is a "pdlaptr" which is a pointer which points to a record "pdlap" where the keys and direction of the overlap region are kept. All of the overlap regions of diffusion and polysilicon represented by this pdlap are also built into a 4-d binary search tree with right thread. "Dnode", "snode", and "gnode" are integer variables to keep track of the node numbers of the drain, source and gate of the transistor. They are initialized to negative integers first. After extraction is complete and all the elements are interconnected, these variables should become positive and indicate the node numbers of the transistors. If the extraction process fails to connect this transistor, the node number will stay negative, and this is very helpful for debugging.

Another useful data structure in the program is a linear list which connects some relevant rectptrs, e.g., the constituting rectangles in an interconnect can be represented as a linked list, and it is defined as :

```
  cntlist = record
```

```
            this        : rectptr;
            next        : cntlistptr;   (* successor in linear list *)
            multipty    : integer;      (* multiplicity *)
            cntdir      : direction;    (* direction of this rect——
                                         used in coupling cap *)
            width       : real;         (* width perpendicular to
                                                 cntdir *)
            found       : Boolean;
          end;
cntlistptr =^cntlist;
```

The parameter "multipty" is used in the computation of interconnect parameters indicating the multiplicity of this cntlist in the search process. "Cntdir" and "width" are designed for use in the computation of coupling capacitances. "Found" is a toggle switch used in the program to indicate the status of this rectangle. It should be mentioned that the creation of this data record to form linked lists of rectangles instead of tagging more fields to the rect record aims at minimizing modifications to the already defined rect record in the original version. This way there may result in some cumbersome algorithms and redundant procedures. Therefore, future improvements on the program may abolish this data record and carefully replan another.

In order to identify all the connected rectangles linked between transistor "ports" which constitute an interconnection system, a data record wire is set up to represent it. A transistor port is defined to be the gate, source, or drain of a transistor. An example of such a wire is shown in Fig. 4.3 with the corresponding equivalent lumped circuit. The record of wire is defined as :

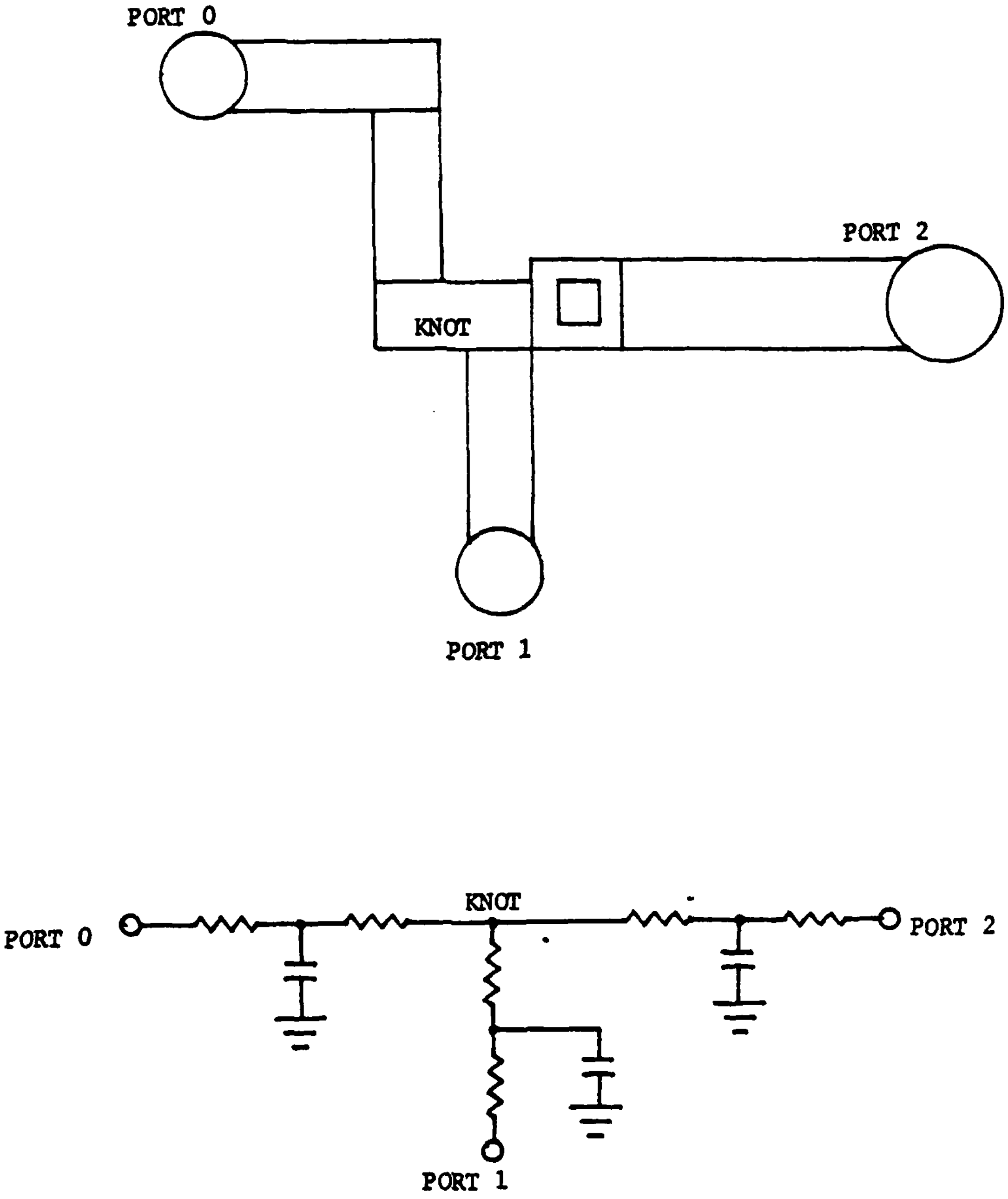


Figure 4.3 An example of a "wire".

```
wire = record
        bound   : array[0..3] of real; (* rectangular bounds *)
        endport : array[0..MXPORT] of portptr;
        portnum : integer;      (* number of portptrs in endport *)
        knotray : array[0..MXPORT] of portptr;
        knotcnt : array[0..MXPORT] of cntlistptr;
        knotnum : integer;      (* no. of elements in knotray,
                                                knotcnt *)
        branchray : array[0..MXPORT] of branchptr;
        branchnum : integer;    (* no. of elements in branchray *)
        wirehead : cntlistptr;  (* head of wire list *)
      end;
wireptr =^wire;
```

"Bound" is the outmost four-corner boundary points of this wire, it is defined the same way as the keys in rect. Its main purpose is to screen off unnecessary comparison between wires in the calculation of coupling capacitances. When two wire bounds are separated by more than the parameter "WIREGAP" specified in the "constant.h" file, no further comparisons between branches inside the wires are needed.

Since a wire links a few transistor ports together at the ends of the wire, a data structure "port" is built to represent those transistor ports. "Endport" is an array which stores those pointers of the transistor ports. A "knot" is a rectangle which is connected to more than two other rectangles in the wire list as shown in Fig. 4.3. In order to conform with the transistor ports at the boundary of the wire, knots are also defined as ports. "Knotray" collects those port pointers pointed to the knots in the wire. "Knotcnt" is also an array of cntlistptrs which point to those knot rectangles. A "branch" is the linked list of cntlistptrs which constitute the interconnection between two ports in the wire. A T-equivalent lumped circuit is specified for each branch in the wire, and the equivalent resistance and capacitance are computed according to the branch. As the others, "branchray" collects those branches. For each array, there has to be an integer to indicate the number of elements in the array, and "portnum", "knotnum" and "branchnum" in the definition are those integers.

4.4. Extraction Procedure

Since the first part of the program which reads in CIF input lines and sets up the 4-d binary search trees for the rectangles on each mask level was obtained from Mike Graf as mentioned earlier, we would like to concentrate on the discussion of the second part of the program, i.e., the extraction of the transistors and the interconnect parameters. The program is coded in PASCAL and separated into nine subprograms and one main program. The first part of the code is contained in subprograms "exfirst.p" and "exsecond.p". Note that "*.p" is the defaulted format on the VAX/UNIX* system to indicate a PASCAL program.

4.4.1. Find the diffusion and polysilicon overlap

First the transistor channel region should be identified. According to the NMOS process in [38], the overlap of diffusion and polysilicon rectangles constitutes a transistor. Therefore, a search for every diffusion rectangle on the mask through the polysilicon binary tree to find any polysilicon rectangle which overlaps this

* VAX is a trademark of Digital Equipment Corporation, and UNIX is a trademark of Bell Laboratories.

diffusion rectangle should be done. If an overlap is found, a record of type pdlap is set up in memory and a pointer to it is generated. Next all those pdlaptrs are stored in a 4-d binary search tree as are those of rectangles. The insertion algorithm will be given in detail in Sec. 4.6 and it is similar to that in [26]. In the record pdlap, the coordinates of the overlap region, the pointers to those two rectangles which form this overlap region are kept.

Note that in the process of finding the overlap region, all the relative positions of two rectangles on a plane have to be considered. They are : overlap, meet at a line, meet at a point, and disconnection. If one rectangle is totally enclosed inside another, this can be considered as a special case of overlap. The cases in which a polysilicon rectangle and a diffusion rectangle meet at a line or at a point are illegal for the design rules employed here. Hence, warning messages will be produced in the log file. Last, the above runctions are performed by the subprogram "exthird.p".

4.4.2. Set up transistors

After the overlap regions are identified, transistor records should be established. Since the channel region of a transistor can be made of connecting overlap regions, they should first be located. For each node in the tree of overlap pointers, a search for adjoining overlap regions in this tree is conducted. All the found connecting overlap regions are built into the gate of a transistor. If no other adjoining overlap regions are found, then this solitary overlap region also constitutes the gate of a transistor. As mentioned before, the possible configurations of a transistor are limited in this extractor for the sake of simplicity, and there can be at most two connecting overlap regions to form a transistor channel.

Next, the drain and the source regions should be identified. They are those remnant regions of diffusion rectangles outside of the channel regions already identified as gates. The above-mentioned limitation is mostly intended to simplify this process. Because the remnant regions from a serpentine-shaped channel region would not be rectangles, but rather polygons, they are not easy to handle within the frame of the present data structure. With the above limitations, the remnants are still rectangles, and they can be separated from the channel regions and identified by a procedure called "split" in subprogram "exfourth.p". Then remrect records are set up for those remnant regions, and all of them are also linked into a 4-d binary tree. In transistor record, the pointers to the source and the drain "remrects" are also kept for future reference in the extractor.

Additionally, the type of the transistor should be defined. Since the "load" transistor is assumed to be depletion type and has its gate connected to its source, the special characteristic of a contact cut overlapped with the gate region is used as a criterion to identify it. The intersection of the ion implant region with the gate region can also be used as the identifying mechanism. However, the former method is employed here. The pointers of those load transistors are kept in an array "loadtran" with an integer variable "loadnum" to indicate its number of elements. The other transistors, be they pass transistors or drive transistors, are kept in another array "othertran". These procedures are performed in subprogram "exfourth.p".

4.4.3. Find the power and the ground

It is assumed that all the ground nodes are connected by a network of rectangles and that they collapse into a single node "0" in the circuit simulation. This applies to the power supply too, and the power network is designated as node "5". In order to string both networks in two linear linked lists, coordinates of two points have to be fed in interactively. Then the two rectangles which encompass these two points are selected as the "heads" of the respective lists. A "depth-first" search [31] is then performed to find all the rectangles in the lists. The procedures "connect" and "setconnect" perform this function. These procedures are used constantly in the latter part of the program to find connecting rectangles for "wires".

The procedure "setconnect" is used to search through the specified mask level given the pointer of a cntlist and its rectangle. The data record cntlist, which was discussed in the previous section, is employed and its pointer is named "cntlistptr". Note also that the parameter lapdicator in the rect record is set for the rectangle which has been selected so that it will not be used again in the search process. The "depth first" search starts with the head cntlistptr and its rectptr as the seed, finds all the rectangles on the given mask level which are connected to it, and links them in the list. It then goes to next cntlistptr on the list, uses it as the seed, and continues to search the same tree for connecting rectangles without a set lapdictor until no rectangles can be found.

Since the constituting rectangles of an interconnect or the power network may not necessarily be all contained on one mask level, the search process should be able to extend to other mask levels as well. Utilizing the fact that those interlevel connections can only be made through contact cuts, the procedure "connect" then performs this extended search. It will call upon "setconnect" and search not only the mask level of the seed but also the mask level of the contact cuts. The pointers of the found contact cuts are also linked in the linear list in the "depth first" search. Then, whenever a contact cut is encountered as a seed in the process, all three levels (diffusion, polysilicon, and metal) have to be searched by the procedure "setconnect". This way all the connecting rectangles, even across different mask levels, can be found, and the power network and the ground network are stored in two lists headed by "powerlist" and "groundlist". The above procedures are contained in the subprogram "exfifth.p".

4.4.4. Set up power and ground nodes

First the remrects of "loadtrans" are searched to see if they are connected with any rectangles in the powerlist. If a contact is found, then this remrect is designated as the drain of the transistor and a default node number 5 is set. The other remrect is automatically set as the source of this load transistor. Next the remrects of " othertrans" are searched and compared with the rectangles in the groundlist. Any remrect connected to this list is set as the source of the transistor and a default node number 0 is designated. Another remrect of this transistor will be set as the drain of the transistor, meanwhile, the transistor will be named as "drive" transistor. The procedures "setpowernode" and "setdrivenode" in subprogram "exsixth.p" perform the above functions.

Note that the resistance and capacitance of the power and the ground networks are not considered in this extractor. The power and the ground are assumed to be at constant voltages and are designated to be node 5 and node 0, respectively. Also the unset nodes will retain their initialized negative values after this step.

4.4.5. Find wires

After extracting the power and the ground network, all the remaining undecided sources, drains and gates of transistors should be linked together by "wires". Starting with the unset source or drain nodes of load transistors, and then the unset source, drain or gate nodes of other transistors, we proceed to find all the wires.

First, using an unset port of a transistor as the head of a list, we can search for connecting rectangles by the procedure "connect", and the data record wire is built in the process. The original starting node of the transistor is the first element in the array "endport" in the wire record. In the process of depth first search, each rectangle in the list is also compared with other unset nodes of other transistors in arrays "loadtran" and "othertran". If a contact is found, then those nodes will be included as ports of this wire in the array "endport". Note that this process only searches for those rectangles which have not been selected in the previous search process. (Lapdicator in the rect record will indicate to that effect.) Thus, the search process will end when all the connected rectangles in an interconnection are found, and the ending ports of the wire will also be found and kept in the array. The above functions are performed by the procedure "findwire" in the subprogram "exeighth2.p".

4.4.6. Compute interconnect resistance and self capacitance

From the data record wire, the equivalent circuit parameters can be computed, i.e., resistance and capacitance. As shown in Fig. 4.2, the interconnection of a wire may be multiple-connected, i.e., there are knots at the intersection of the branches. In order to identify those knots, the linked list of rectangles has to be searched first. Any rectangle in the list having more than two connections to other rectangles is considered a knot and they are stored in the arrays "knotray" and "knotcnt". A branch is defined as a linked list of rectangles between two ports remembering that knots are also ports. For each branch in the wire, a corresponding T equivalent lumped circuit is set up as shown in Fig. 4.2. Heuristically the two resistances are assumed equal and half the value of the dc resistance which is calculated from the linked list. Furthermore, the capacitance to ground is the self-capacitance calculated by the "cylindrical approximation formula" given in Chapter 3. Note that the formula is a two-dimensional one, and the calculated value from the formula has to be multiplied by the length of the branch for the total capacitance value. No special considerations are given to corners and bends in the branches.

It is important to mention that the rectangles of the knots are included in all parameter computations of the branches which are connected to them. If a long rectangle is designated as a knot, then the calculated parameters of the branches may be too large. In other words, this is a pessimistic estimation of the parameters. It is hoped that this situation may be alleviated by careful layout or by developing further procedures to recognize it and split the rectangles to make it congruous to the calculation. Some of the procedures in the subprograms "exseventh2.p" and "exeighth2.p" perform the above tasks.

4.4.7. Set up node numbers

Before reporting the calculated electrical parameters, we have to first check them with the criteria in file "constant.h". There are a few cases to consider : (i) If both the capacitance and the resistance values are larger than the criteria, "CLIMIT" and "RLIMIT", all components of the T equivalent circuit are reported. (ii) If only the resistance is smaller than "RLIMIT", then the node numbers at the endports of this branch should be the same, and the capacitance should be inserted between this node and the ground. Furthermore, if the capacitance is also smaller than "CLIMIT", just assign the two connecting endports with the same node number and forget the T equivalent circuit. (iii) If only the capacitance is smaller than "CLIMIT", no center node is needed in T circuit, and the two resistors are collapsed into one resistor connecting the two endports.

Since knots are considered as ports, node numbers are assigned to knots also. Generally, in determining the node numbers at the end of resistors, first check the connecting ports. If node numbers have been assigned to the connecting endports, then the resistor will use those node numbers at its ends. Otherwise, new node numbers are generated. The center nodes where capacitors are connected to ground are numbered as 5xx, so that they can be easily recognized.

4.4.8. Compute coupling capacitances

In the computation mentioned above, only the self-capacitances of the interconnects are considered. To further compute the coupling capacitances, the proximity of the "wires" has to be checked. First, the bounds of the wires are compared pairwise. If they are disconnected by a margin larger than "WIREGAP" specified in the file "constant.h", then no further comparison will be made to avoid possible confusions. Otherwise, the branches in both wires have to be compared pairwise. In this check, another parameter "BRANCHGAP" in the "constant.h" file is used to test.the separation of the branches. Only when the separation between two interconnects is smaller than this parameter is the coupling capacitance between them computed by the approximate formula in Chapter 3. Since there are quite a few different scenarios of two close interconnects, e.g., they can be parallel on the same level or on different levels, or a metal line can cross over a polysilicon line perpendicularly, etc., every case of two close branches has to be considered individually. Also these scenarios may be dependent upon the technc'ogy employed and the parameters of the fabrication processes. In the present version of the extractor, silicon NMOS technology is assumed and only parallel lines on the same level and the perpendicular cross-over are considered in the coupling calculation. In Fig. 4.4, an example of two parallel interconnects is shown. Note that the capacitance computed by Eq. (3.12) is capacitance per-unit-length; thus, it must be multiplied by the total length in question. In the case shown in Fig. 4.4, the length used in the computation is l_1+l_2. This apparently is less than the actual length because of the omission of the turn. The T equivalent circuits for these two interconnects are also shown in Fig. 4.4. One of the advantages of using the T circuit is that the coupling capacitances can be easily inserted between the center nodes of the two T circuits of the adjacent interconnects. The procedures "sortwire" and "comparewire" in the subprogram "exninth2.p" perform the above functions. It should also be mentioned that the self-capacitances of an interconnect will be reduced slightly due to another interconnect in the vicinity as discussed in Sec. 3.3.1. However, this effect is not taken into account in the computation of the self-capacitances in this extractor.

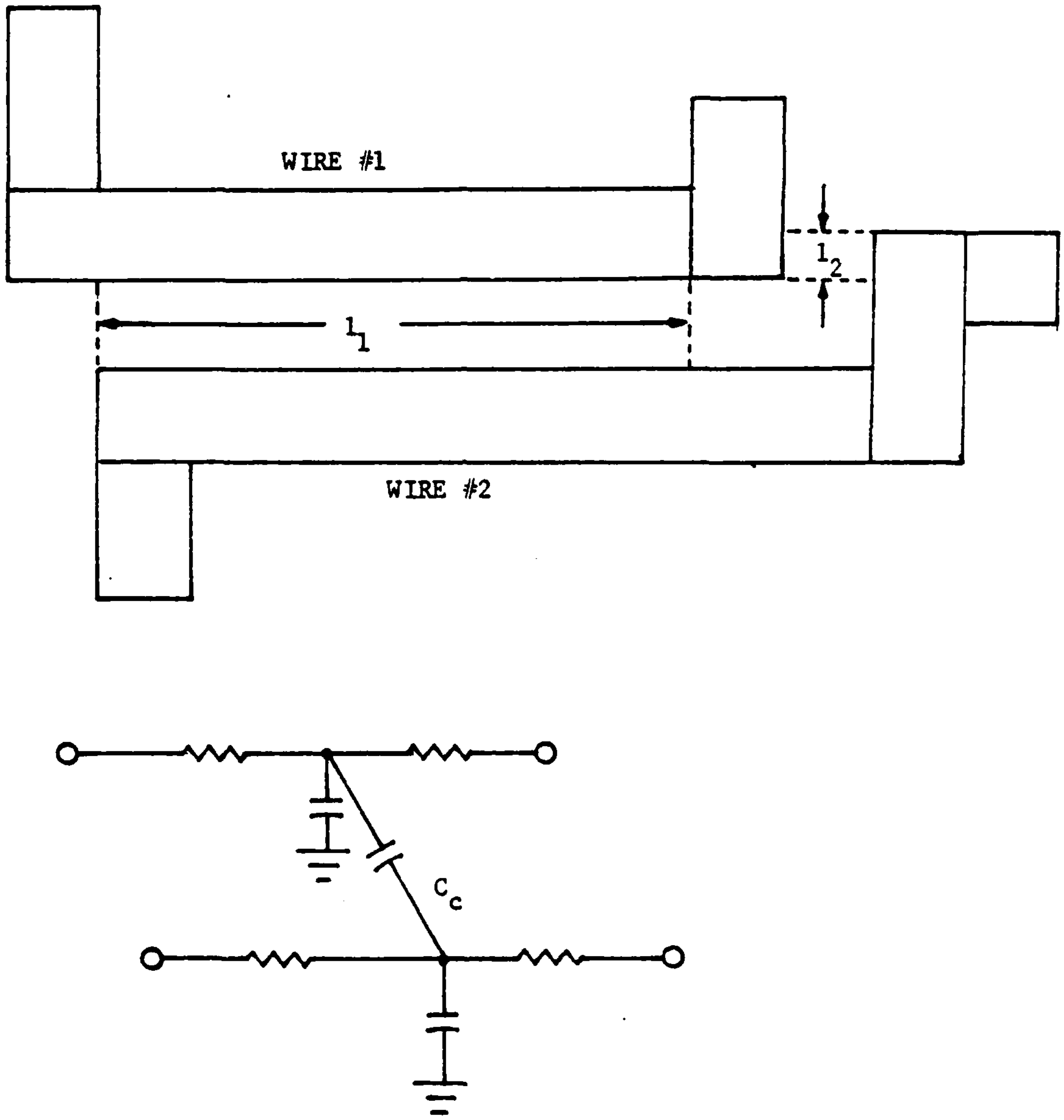


Figure 4.4 Consideration for the evaluation of coupling capacitance between two parallel interconnects.

4.4.9. Report

Basically, The extractor generates two output files. One is the log file and the other is the SPICE file. The log file records warning and error messages and all the steps taken in the extraction process along with the time and date for reference. The original CIF input file and some data structures like wires and transistors are also included for debugging. It is defaulted to "list.out" or another name given by the user. A detailed explanation of the usage and commands of the extractor can be found in the procedure header of "getcmdlnargs" in the subprogram "exsecond.p".

Another file is the SPICE file which includes all the parameters of the transistors and the interconnect resistances and capacitances. The format on this file is compatible with the SPICE input format. The node numbers are set in the extraction process and cannot be set externally. This may need to change in the future version so that the user can set the crucial node number to his choice to improve the readability of the output. This file is defaulted to the user's terminal, if not otherwise redirected or renamed. A few examples will be discussed in the next section and some SPICE files are shown there.

4.5. Examples and Results

In this section, a few examples to illustrate the network extractor are presented. The circuit layouts are drawn manually employing the design rules comformable to those in [38]. After supplemented with some model cards and input output specifications, the extracted outputs of the circuits are fed into SPICE, and the simulated circuit responses are compared.

4.5.1. Example I : Static RAM cell

In Fig. 4.5, the layout of a six-transistor static RAM cell is shown. After extraction by the network extractor, the output SPICE file is given in Table 4.1. "C000", "r000" and "rr000" are the three components in a T equivalent circuit for an interconnect. "R000" and "rr000" have the same value of resistance. The center node of the T circuit where the capacitors are connected to the ground is numbered from 500. The capacitance "c003" is the only coupling capacitance between the interconnects of nodes 7,10 and nodes 6,11. That is, the coupling capacitance of the metal and polysilicon cross-over at the center of the layout.

A schematic circuit diagram of the circuit is also shown in Fig. 4.6. Note that the negative node numbers on "mot000" and "mot002" are due to the dangling nodes connected to the bit line and the word line on the pass transistors at both ends. Those negative nodes are helpful in locating not only the input and output nodes, but also the erroneous disconnected nodes. It should also be mentioned that, in the extraction process, the parameters "RLIMIT" and "CLIMIT" in the file "constant.h" are specified as zero, so all of the resistances and capacitances of the interconnects are computed and listed in Table 4.1.

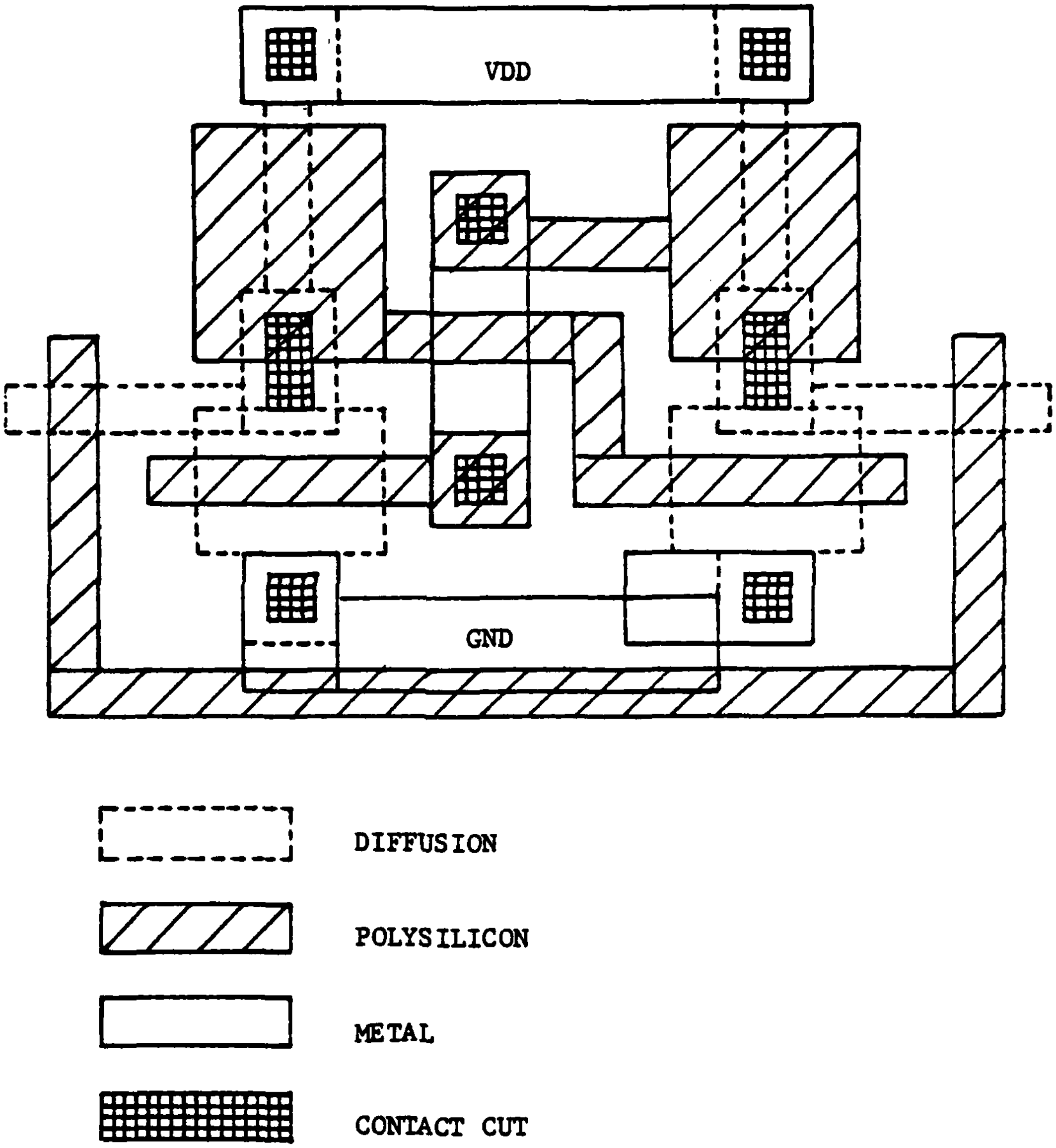


Figure 4.5 Layout of a six-transistor SRAM cell.

TABLE 4.1 SPICE Output Listing for SRAM Cell.

```
c000    500   0   0.0110pf
r000      8 500   2.9e+02
rr000   500   9    2.9e+02
c001    501   0   0.0041pf
r001     10 501   1.1e+02
rr001   501   7    1.1e+02
c002    502   0   0.0077pf
r002     11 502   7.5e+01
rr002   502   6    7.5e+01
c003    501    502   0.0020pf
mld000      5   6   6  0   load     l=  9.00um     w=  2.67um
mld001      5   7   7  0   load     l=  9.00um     w=  2.67um
mot000      7   8  -1  0   pass     l=  2.00um     w=  2.00um
mot001      6  10   0  0   drive    l=  2.00um     w=  8.00um
mot002     -2   9   6  0   pass     l=  2.00um     w=  2.00um
mot003      7  11   0  0   drive    l=  2.00um     w=  8.00um
```

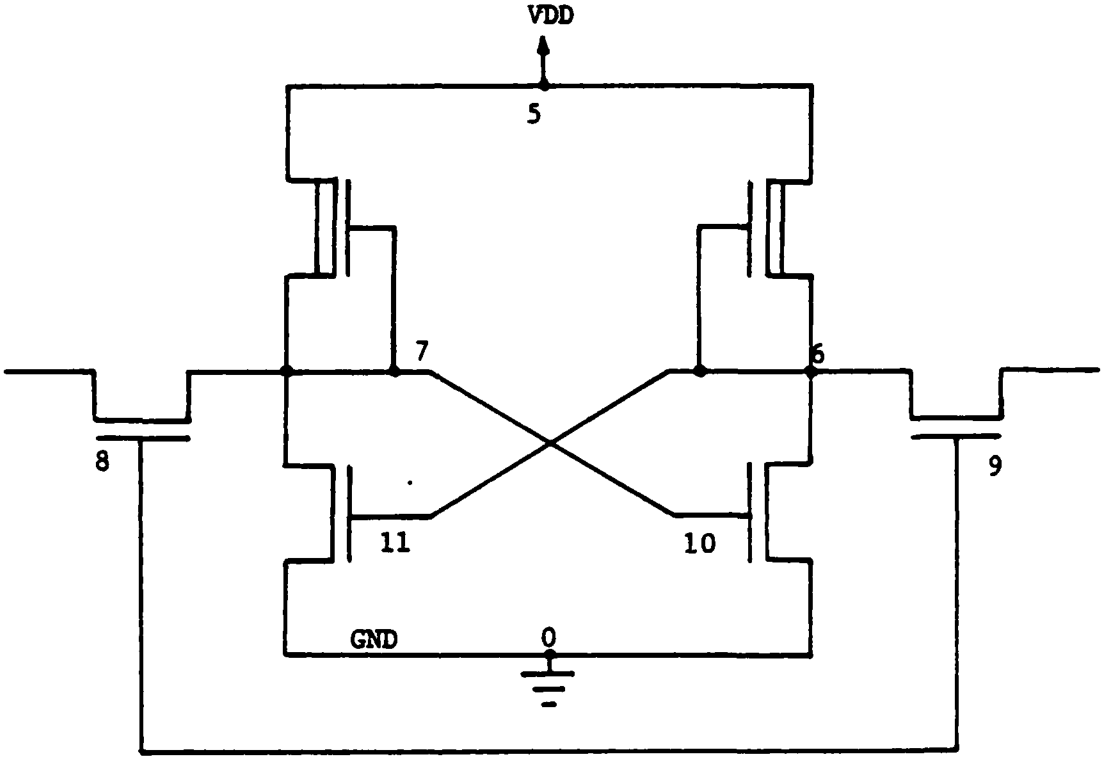


Figure 4.6 Schematic diagram of SRAM cell.

4.5.2. Example II : Full Adder Circuit I

The schematic circuit diagram of "Full Adder Circuit I" is shown in Fig. 4.7. With the same specifications in the "constant.h" file as in the previous example, we can extract the SPICE file from the extractor. Combined with some model cards, input and output specifications, and a few modifications to the circuit, a few simulations* of the circuit are performed. In order to distinguish the effects of the interconnect parameters, three cases are considered, namely : (i) the circuit without taking into account the interconnect parameters; (ii) the circuit with only the self-capacitance and the resistance of the interconnects; (iii) the circuit with all interconnect parameters including the coupling capacitances. These three simulated responses of "Full Adder Circuit I" are shown in Fig. 4.8 to Fig. 4.10. The full SPICE deck for case (iii) is given in Appendix E. It should be pointed out that all eight possible combinations of the three inputs to the full adder are covered in the simulations, and only the first input waveform is shown in the graphs. The other two input waveforms are not plotted for the sake of neatness of the graph.

In comparing Fig. 4.9 with Fig. 4.8, it is obvious that the rise and the fall times in Fig. 4.9 are larger than those in Fig. 4.8. This is naturally due to the RC time delay contributed by the interconnects in the circuits. It should be mentioned that the time

* These simulations are actually done by SLATE [48,49] which has a similar format as SPICE and does a better job at transient analysis.

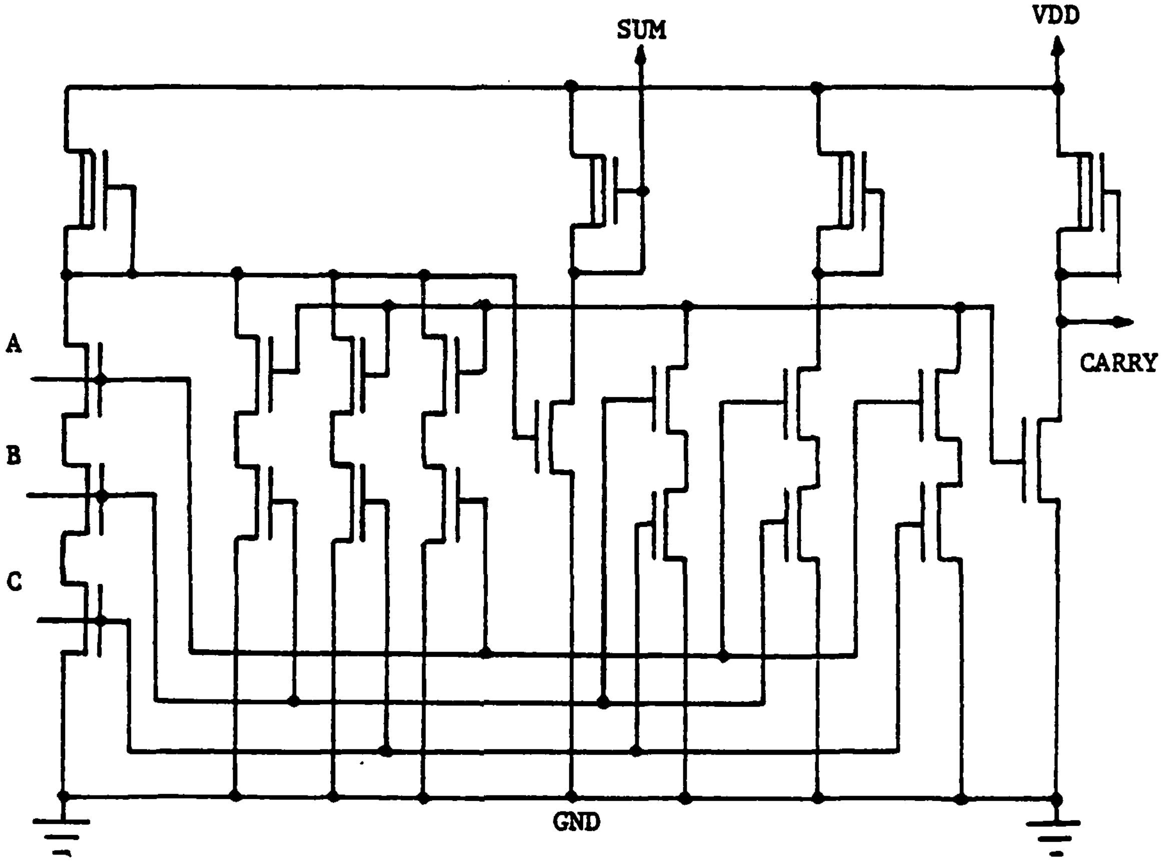


Figure 4.7 Schematic diagram of Full Adder I.

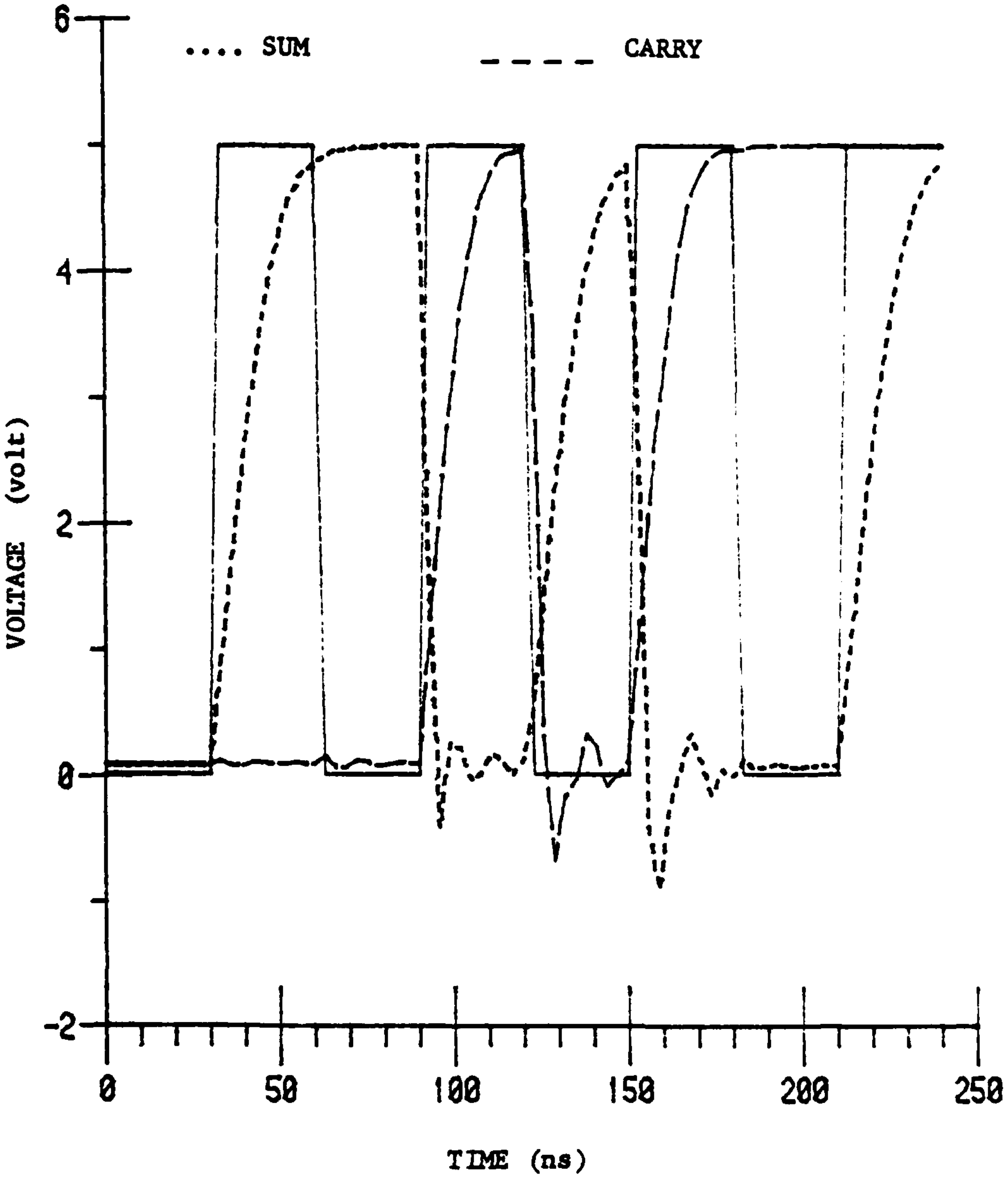


Figure 4.8 Full Adder Circuit I with no interconnect parameters.

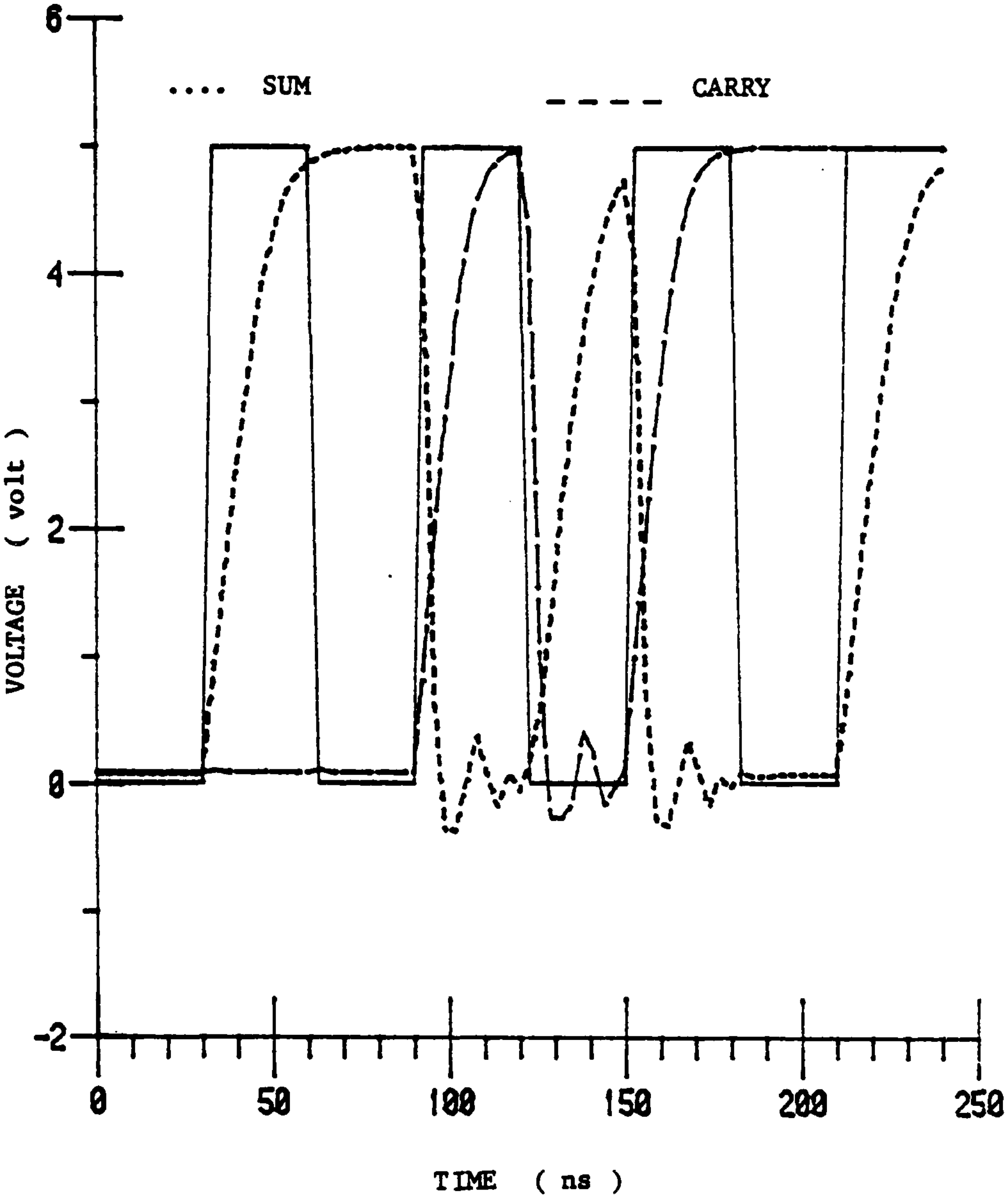


Figure 4.9 Full Adder Circuit I with self-capacitance.

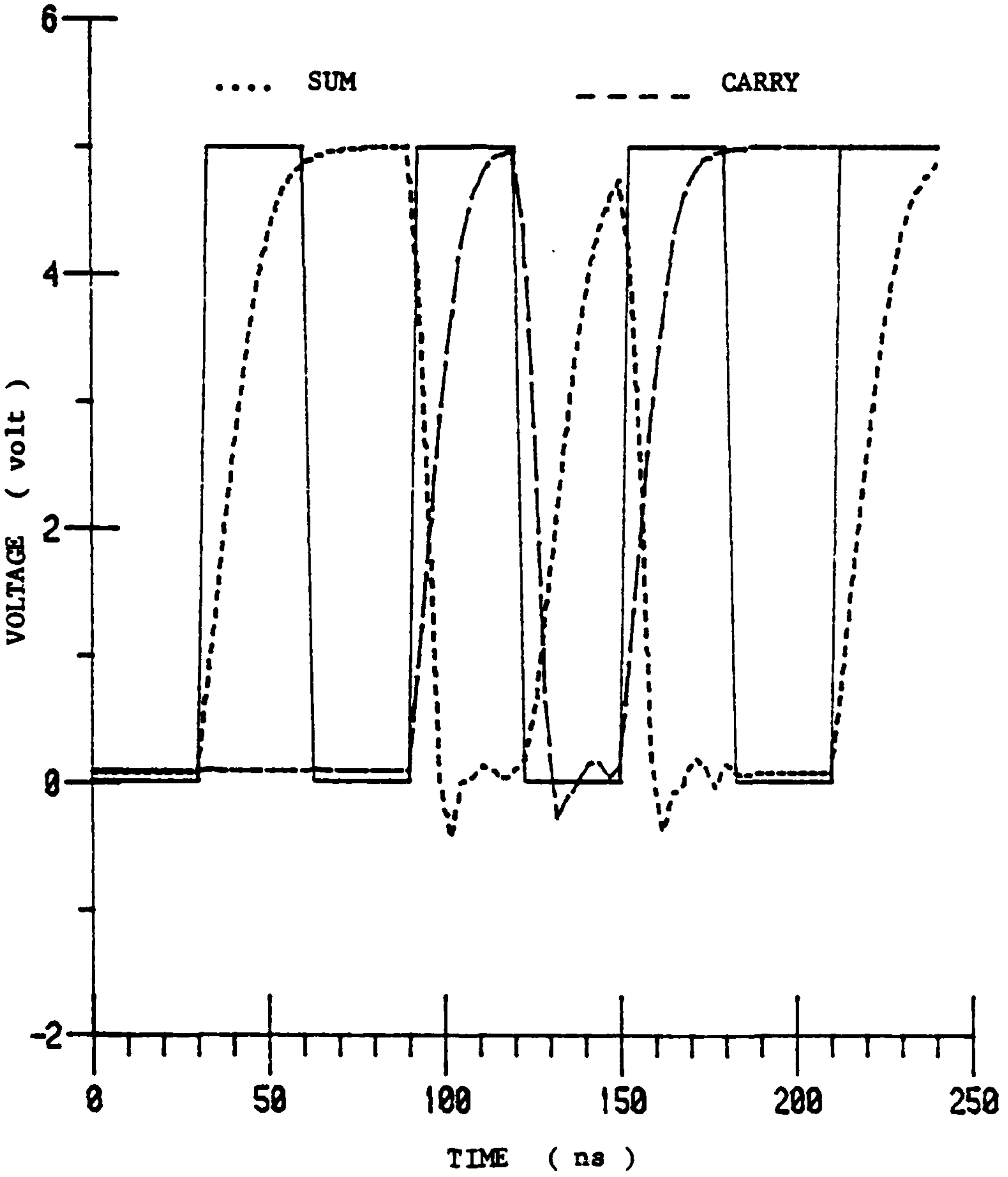


Figure 4.10 Full Adder Circuit I with all parameters.

delays shown in Fig. 4.8 may be attributed to the capacitive loads of 0.1 pF at both output ports. Furthermore, these loads are used throughout all the circuits discussed in these two examples of full adder circuits.

The circuit response of the third case which also includes the coupling capacitances are shown in Fig. 4.10. They differ little from the responses shown in Fig. 4.9. On close examination, there may be just a little more delay. Overall, it can be said that the coupling capacitances for this circuit do not influence the circuit responses much. However, the inclusion of the self-capacitances and the resistances has more visible effects on the responses, and they should be considered in the circuit simulation.

4.5.3. Example III : Full Adder Circuit II

Another full adder circuit is used as an example. Its schematic is shown in Fig. 4.11. There is a basic design difference from the previous full adder circuit. In this case, more pass transistors are used. Noticeably, they are used at the output ports; thus, the output responses of this circuit cannot reach the full 5 volts of VDD. At most, the logic "1" for the outputs can reach 5 volts minus the threshold voltages of the pass transistors which are set at 1 volt. This point is manifested in the following graphs. Those three cases mentioned in the previous example are also considered here. In Fig. 4.12, the responses of "Full Adder Circuit II" without the interconnect parameters are shown. Figure 4.13 is for the case of taking the self-capacitances and the resistances into account.

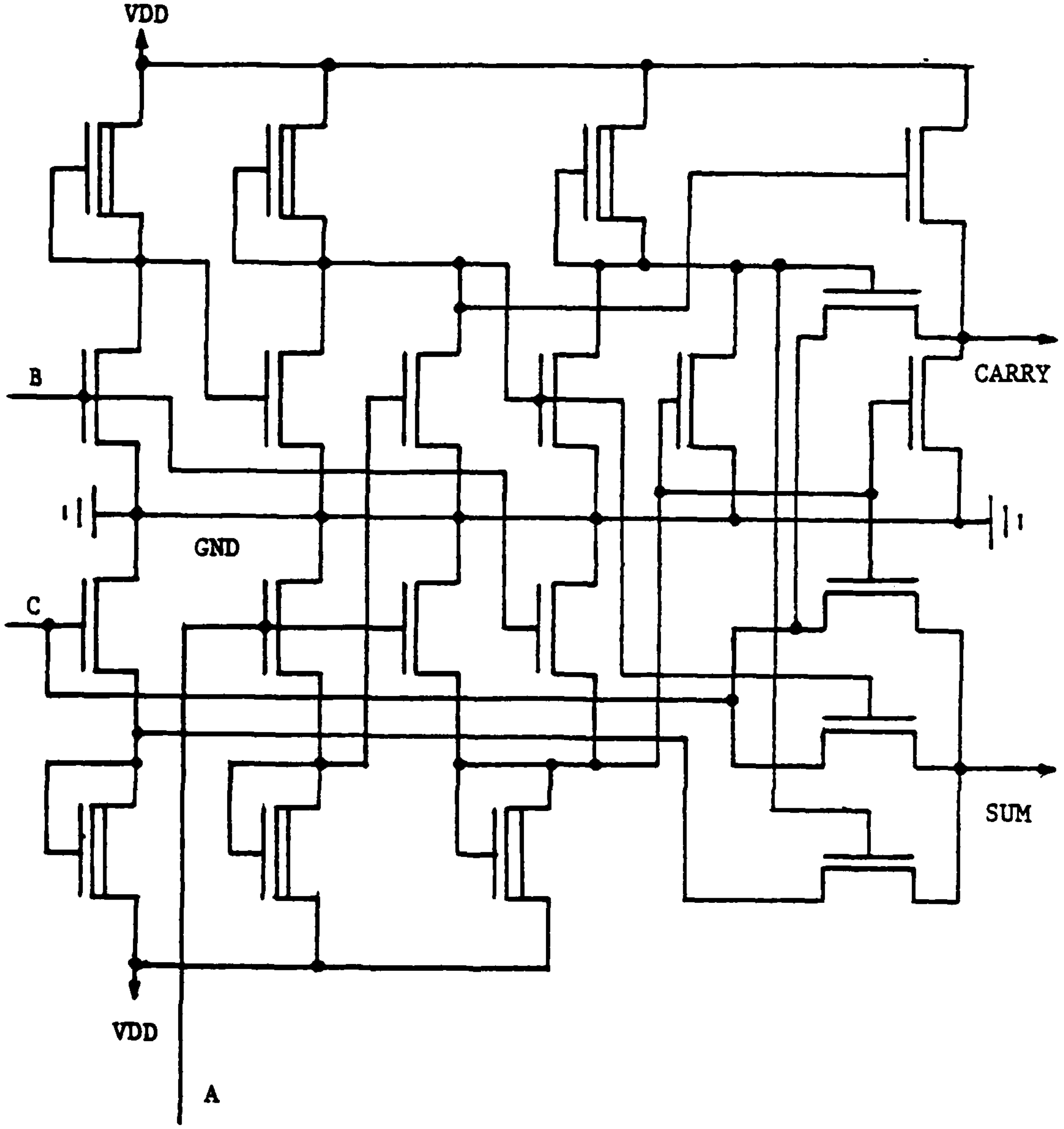


Figure 4.11 Schematic diagram of Full Adder Circuit II.

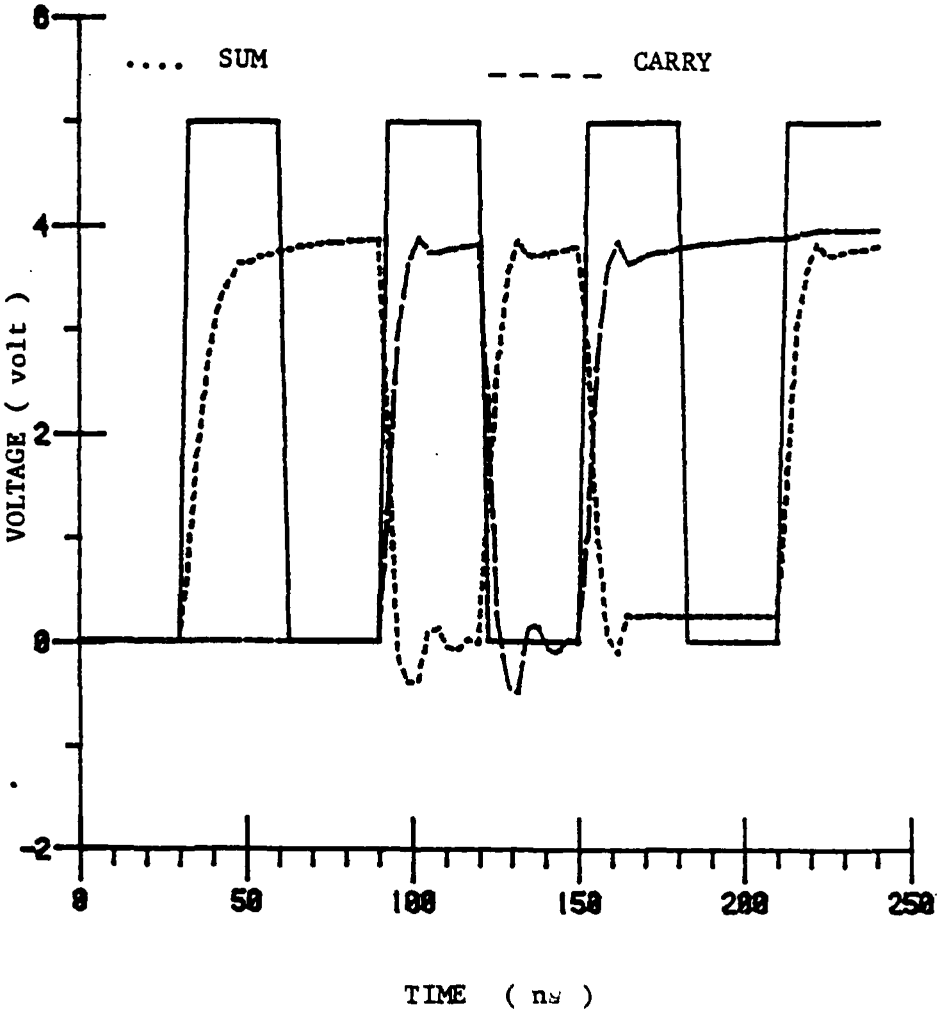


Figure 4.12 Full Adder Circuit II with no interconnect parameters.

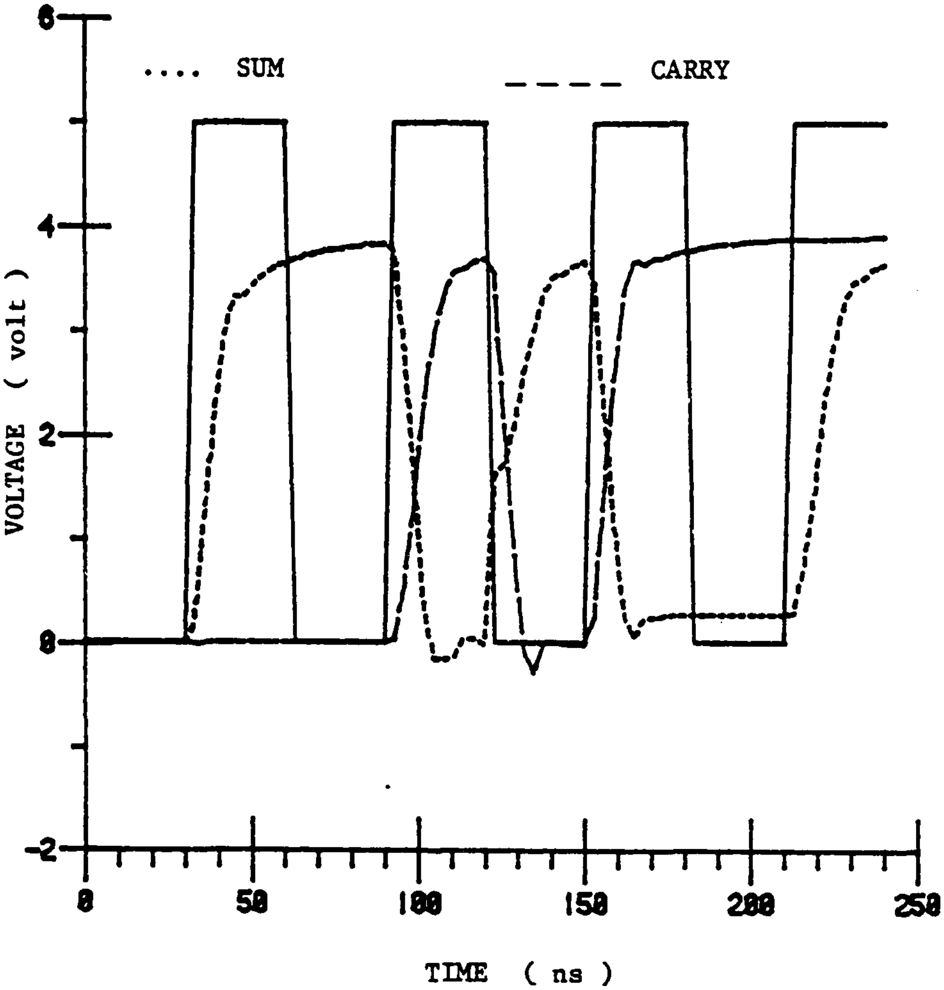


Figure 4.13 Full Adder Circit II with self-capacitances.

Finally, in Fig. 4.14 the responses with all the interconnect parameters included are considered. The "constant.h" file, the model cards, and the load capacitances are chosen the same as in Example II; the listing of the SPICE input deck is given in Appendix F.

In comparing the responses* for these three cases, it is obvious that the coupling capacitances in this circuit have greatly affected the output responses. Especially, the waveforms in Fig. 4.14 differ drastically from those in Fig. 4.12 and Fig. 4.13. Not only are the rise- and the fall-time delays of the responses larger than those in Fig. 4.13 and Fig. 4.12, but also there is a notch developed in the sum waveform. This may result in unwanted errors or hazards for the circuit. Hence, the circuit or the layout of the circuit should be examined more, or redesigned, to correct this possible flaw. In design verification, it is our aim to discover and eliminate those possible errors. On the other hand, the responses in Fig. 4.13 with only the self-capacitances considered differ from those in Fig. 4.12 by larger rise- and fall-time delays as the case in the previous example.

To sum up, it can be concluded that the coupling capacitances of the interconnects are important in the circuit simulation in this example. However, in the previous example, they had an insignificant effect on the circuit responses. Consequently, coupling capacitance effects depend heavily on the mask layout and the circuit itself.

* These simulations are also done by SLATE.

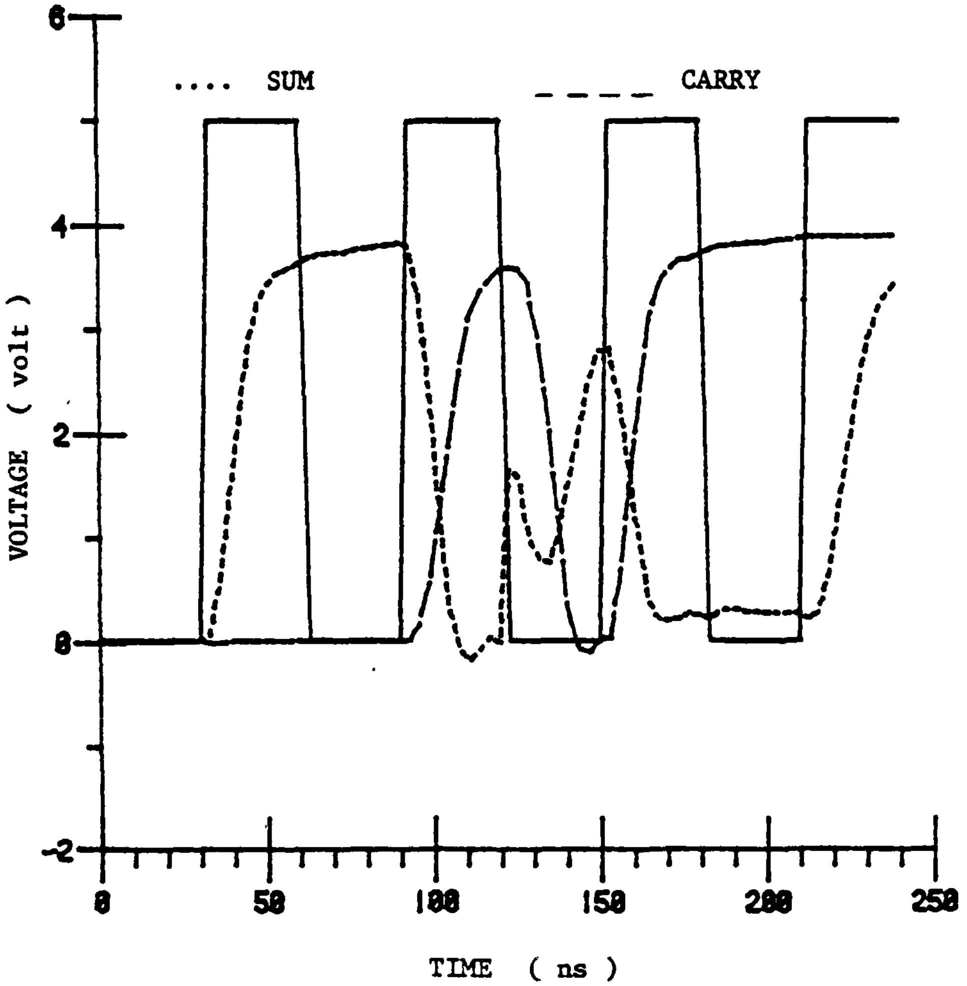


Figure 4.14 Full Adder Circuit II with all parameters.

Their effects on the circuit have to be investigated individually. As to the self-capacitances and the resistances of the interconnects, they surely contribute to the time delay of the circuit and should be considered.

4.6. Computational Complexity of the Extractor

Because of the increased circuit complexity and myriads of data involved in VLSI circuits, it is essential that the CAD tools consist of computationally efficient algorithms. The insertion algorithm of the 4-d binary search tree with right thread, which is the basic data structure of the extractor, is given partly in [26] and listed as follows :

```
Given a node P to be inserted into the tree.

I1. [Check for empty tree]  If ROOT = nil, then ROOT <- P,
    RSON(P) <- nil, LSON(P) <- nil, set RSON(P) as a thread,
    return;  otherwise, Q <- ROOT.

I2. [Comparison]  If $K_i(P) = K_i(Q)$, $0 \leq i \leq 3$, and
    i is the discriminator, then return;  otherwise, set SON(Q).
    (RSON or LSON depending on the comparison of Key(P) and
    Key(Q)).  If SON(Q) = nil, then goto I4.

I3. [Move down the tree]  Set Q <- SON(Q), goto I2.

I4. [Insert new node in tree]  Set SON(Q) <- P, RSON (P) <- nil,
    LSON(P) <- nil, return.
```

A detailed discussion of the complexity of the above algorithm was given in [26], and it was concluded that typical insertions and record look-ups in a 4-d tree will examine approximately $1.386\log_2 n$ nodes, where n is the size of the tree. In the extractor, this insertion algorithm is called to set up a 4-d binary search tree for rectangles on each mask. It is also set up for the diffusion and polysilicon overlap rectangles.

Next in search of the diffusion and polysilicon overlap regions, a search algorithm is needed. For example, given a polysilicon rectangle P with its four keys, we have to find all the diffusion rectangles Q which are overlapped with P. In other words, the keys of P and Q have to satisfy all the following conditions simultaneously :

$K_0(Q) \leq -K_2(P)$

$K_1(Q) \leq -K_3(P)$

$K_2(Q) \leq -K_0(P)$

$K_3(Q) \leq -K_1(P)$

This search is classified as "Intersection Queries" in [26,30]. The search algorithm can be described as follows :

```
Given a rectangle P, try to find all the overlapping rectangles Q's
in the tree rooted by ROOT.

I1. [Initialization]  Q <- ROOT.

I2. [Move to leftmost corner]  While LSON(Q) $\neq$ nil do
    Q <- LSON(Q) and INCREMENT(i), i is the discriminator.

I3. [Check intersection]  If $K_j(Q) \leq - K_{(j+2)\mathrm{mod}4}(P)$,
    $0 \leq j \leq 3$, then report Q, continue I4.

I4. [Go to predecessor]  If $K_i(Q) > - K_{(i+2)\mathrm{mod}4}(P)$ or
    RSON(Q) is a thread, then set Q <- PREDECESSOR(Q).
    If Q = nil, then return;  otherwise goto I2.
```

The above algorithm is similar to that described in [27]; basically, it is an "inorder" traverse of the binary search tree. The main advantage that this data structure offers in saving traverse time is due to the check on step I4. If the condition $K_i(Q) > -K_{(i+2)mod4}(P)$ is satisfied, then the right subtree of Q need not be traversed, and the search can be directly positioned to the predecessor of Q. Note that only one condition is checked in this step, and i is the discriminator. Instead, all four keys have to be examined in step I3. Moreover, the precedure to find the predecessor can be accomplished rather easily with the help of the right threads woven in the tree instead of employing a stack register to store the nodes on the way down the tree as suggested in [27]. Hence, this is another saving in the search time. However, due to this step in the search algorithm and the random nature of the tree, it is difficult to evaluate the computational complexity for this algorithm. Some empirical data had been collected to test the complexity as discussed in [27]. It may be inferred that the search algorithm of the 4-d binary tree is rather efficient.

Other possible expensive algorithms in the extractor may be those for establishing the interconnect data structure wire and finding the coupling capacitances. Because the transistors are stored in linear arrays, a search through the whole arrays has to be done for each interconnect rectangle to check if it is connected to the transistor ports. Also, the algorithm of finding the coupling capacitances is done by first comparing the wires pairwise, though "wirebounds" are used first to screen off far-away candidates; then, all the branches inside the wires are compared pairwise. Both of these algorithms are approximately n^2 processes, and they can be computationally intensive. Thus, they are in want of some improvements in the future versions.

In order to experimentally estimate the computational complexity of the extractor, a few cases of a chain of "Full Adder Circuit II", presented in the previous section, are tested. The number of transistors in the circuit is used as a measure of the complexity of the circuit. The results are given in Table 4.1. An empirical relation between the number of transistors (N) and CPU time (T) can be assumed as : $T = \alpha N^{\beta}$. From the data in Table 4.1, the exponent β can be estimated to be in the range of 1.3 to 1.7. Therefore, the computational complexity of the extractor is proportional to $O(N^{1.3})$ to $O(N^{1.7})$, where N is the total number of transistors. This is

TABLE 4.1 CPU Time Consumption of the Extractor.

no. of transistor (N)	CPU Time (T) in sec
21*	13.3
42	33
63	58.1
84	84
105	122.4
168	264
336	851.2

* One "Full Adder Circuit II" contains 21 transistors.

comparable to some other extractors, e.g., the extractor discussed in [28] has complexity of $O(N^{1.8})$.

In summary, from the examples shown in this chapter, it can be concluded that the inclusion of interconnect parameters in the circuit simulation is important. One can discover possible timing errors in the design or the layout. In order to handle more complex circuitry, the algorithms employed in the extractor need to be efficient, and hierarchy should be built into the program.

CHAPTER 5

CONCLUSION

The complexity of VLSI circuits has necessitated the development of more comprehensive CAD systems. One of the problems addressed in this thesis is concerned with the interconnection circuit. It is essential to take into account the electrical parameters of the interconnects in the more complicated VLSI circuits as they gradually become the dominating factors. Furthermore, it should also be "computationally feasible" to extract this information from the layout in the design verification. Hence, there are basically two parts in this thesis. The first part is concerned with the accurate computation of the interconnect capacitances with numerical methods, more specifically, the integral method was employed here. Then simple formulas for the interconnect capacitances are developed in the thesis with a view to easily incorporate them into an extractor. The second part of the thesis deals with the extraction of interconnect parameters, and a network extractor was developed for it.

In Chapter 2 a detailed formulation of the integral method was given. It encompasses the derivation of the Green's functions in homogeneous, two layers and three layers of media; the even-odd mode analysis for multi-conductor; and the use of the "method of moments" to solve the integral equation. Pulse-type functions were used for both the basis functions and the testing functions for the sake of closed-form evaluation of kernel matrix elements. This method is integrated into two FORTRAN programs "CAP2D" and "CAP3D". All the self capacitances with respect to the ground of every interconnect in question and all the coupling capacitances between interconnects can be evaluated up to three layers of media and at most ten interconnects. For the two-dimensional case, slant side walls of the interconnects were also considered. As the examples in Chapter 2 and Appendix B show, a variety of interconnect capacitances can be evaluated by these programs, and they are reasonably efficient and versatile. However, in three layers of media, the Green's function has become quite complicated and the computation may be intensive. If detailed capacitance information is needed for irregularly shaped boundaries, the integral method may not be suitable.

As for the simple formulas, the "cylindrical approximation formula" was derived in Chapter 3 for the self-capacitance. It is based on the known formula of the capacitance of a cylinder above a ground plane. As shown in Chapter 3, it compares favorably with other simple formulas and is reasonably accurate with respect to the numerical results. Then a "least square fit" method was employed to find simple formulas for the coupling capacitances between two parallel interconnects or two interconnects on different levels which cross over each other. These prove to be useful in the network extractor. As the interconnect structure becomes more complicated, simpler formulas may be necessary to include all the capacitances in the extraction.

Finally, a detailed description of the network extractor was given in Chapter 4. It is a rectangle-based, Manhattan-geometry-only, NMOS extractor. It reads in CIF input of the layout information, then puts out circuit components in SPICE compatible form. The SPICE output file includes transistors and their parameters, e.g., channel width and channel length and the resistance and capacitance of the T-equivalent lumped circuit of the interconnect. A few examples were given in Chapter 4, along with some circuit simulations. From these examples, it is apparent that the interconnect parameters certainly play an important role in the circuit simulation. Although the original design did not emphasize the efficiency of the extractor, it was found out by experiment that the computational complexity of the extractor is approximately proportional to $O(N^{1.7})$, where N is the number of transistors in the circuit. However, due to simplification of the transistor extraction, only a limited repertoire of transistor configurations is recognized in the extractor. Also, some redundancy in the data structure and possible inefficiencies in the algorithms may need further improvements. In all, it fulfills the need to have the interconnect parameters extracted and utilizes the simple formulas derived in Chapter 3. Besides increasing the repertoire of transistor configurations recognizable to the extractor and improving some of the algorithms and data structures in the program, it is also necessary to link the extractor hierarchically with other CAD tools to handle larger circuits.

APPENDIX A

THE VALIDITY OF THE LUMPED CIRCUIT MODEL

A.1. Introduction

The lumped circuit model for the interconnect is employed in this thesis. A discussion is presented here to establish the validity of this approach through the transmission line theory.

A.2. Formulation

Consider a transmission line circuit and its T-equivalent circuit shown in Fig. A.1 and Fig. A.2. The length of the line is "1" and "I_s" and "Z_1" are the current source and the line load, respectively. The characteristic impedance and the propagation constant are :

$$Z_o = [(R+j\omega L)/(G+j\omega C)]^{1/2}$$
$$\gamma = [(R+j\omega L)\cdot(G+j\omega C)]^{1/2}$$

where R, L, G, C are the resistance, inductance, conductance and capacitance per unit length of the line. From the transmission line theory, the input impedance Z_s and the voltage response at the load end of the line V_1 can be expressed as [46]

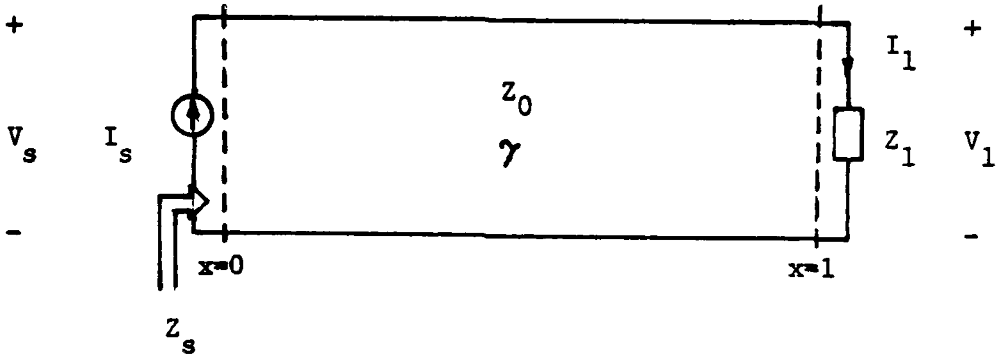


Figure A.1 Transmission line circuit.

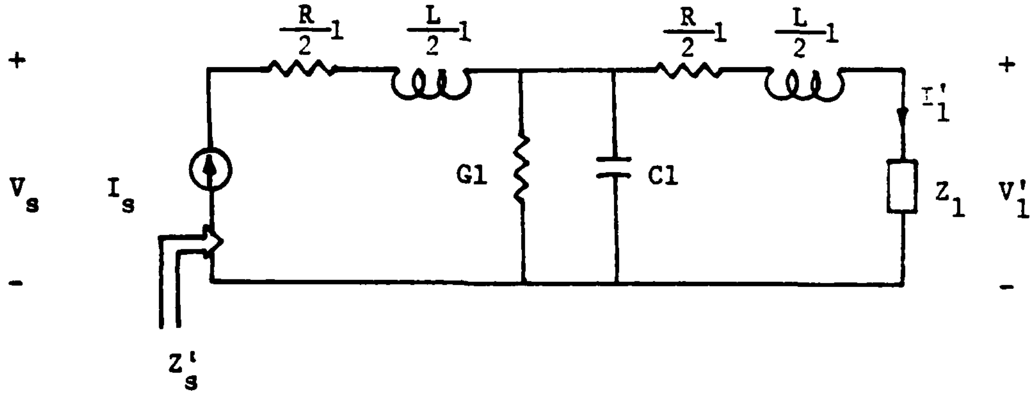


Figure A.2 Equivalent lumped circuit.

$$Z_s = Z_0 \frac{Z_l\cosh(\gamma l) + Z_0\sinh(\gamma l)}{Z_l\sinh(\gamma l) + Z_0\cosh(\gamma l)} \qquad (A.1)$$

$$V_l = \frac{I_s Z_0 Z_l}{Z_l\sinh(\gamma l) + Z_0\cosh(\gamma l)} \qquad (A.2)$$

These two parameters define the characteristics of the transmission line, and they will be compared with those of the "equivalent" lumped circuit model.

Next consider the T-equivalent lumped circuit in Fig. A.2. All the parameters are defined the same as those in Fig. A.1. Then the input impedance Z_s' and the voltage response V_l' of this lumped circuit can be easily obtained as

$$Z_s' = Z_0 \frac{Z_l(1+0.5\gamma^2 l^2) + Z_0(\gamma l+0.25\gamma^3 l^3)}{Z_0(1+0.5\gamma^2 l^2) + Z_l(\gamma l)} \qquad (A.3)$$

$$V_l' = \frac{I_s Z_0 Z_l}{Z_0(1+0.5\gamma^2 l^2) + Z_l(\gamma l)} \qquad (A.4)$$

In order to show that the lumped circuit in Fig. A.2 is an "equivalent" circuit to the transmission line circuit in Fig. A.1, the input impedance and the voltage response should be correspondingly equal. In other words, Eq. (A.3) should be equal to to

Eq. (A.1) and Eq. (A.4) to Eq. (A.2). Note that if the hyperbolic functions, sinh and cosh, in Eq. (A.1) and Eq. (A.2) are expanded in a Taylor series, the first-order approximations are :

$$\sinh(\gamma l) \simeq \gamma l + \ldots.$$
$$\cosh(\gamma l) \simeq 1.0 + 0.5\gamma^2 l^2 + \ldots.,$$

Furthermore, if the above approximations are substituted into Eq. (A.1) and Eq. (A.2), Eq. (A.3) and Eq. (A.4) are obtained except for the term $0.25\gamma^3 l^3$ in Eq. (A.3). Therefore, it may be concluded that the equivalent lumped circuit is valid, as long as the parameter γl is small enough.

A.3. Discussion

A.3.1. Lossless transmission line First, let us consider a lossless transmission line. In this case, the resistance and the conductance of the line are relatively small, i.e., $R \ll \omega L$, and $G \ll \omega C$. Then the propagation constant becomes

$$\gamma = j\beta = j\frac{2\pi}{\lambda}$$

where λ is the wavelength of the propagating wave on the transmission line. In order to satisfy the criterion arrived earlier, i.e., the factor $\gamma l = j2\pi l/\lambda$ must be small, the length of the line "l" should be small compared with the wavelength on the line λ. This is the usual criterion for the validity of the lumped circuit model.

A.3.2. RC lossy line

Next, consider a lossy line with negligible inductance and conductance, i.e., R >> ωL, and G << ωC. In this case the propagation constant becomes

$$\gamma = (1+j).(\frac{\omega RC}{2})^{1/2}$$

To keep γl small as discussed before, the factor $(\omega RC)^{1/2}l$ should be kept small too. In order words, $\omega^2 RCl^2$ should be small. The term RCl^2, as a whole, can be considered as the "time delay" of the transmission line, and the factor l^2 is due to the per unit length basis of the resistance and the capacitance. If T is the period of the propagating wave, and $\omega = 2\pi/T$, then $\omega RCl^2 = 2\pi(RCl^2/T)$. The criterion for the validity of the lumped circuit, i.e., small γl, can be interpreted in this case as implying that the delay of the line should be small compared with the period of the wave. For a typical example [47], a doped polysilicon line with 30Ω/square and 1 cm long has a time delay in the order of 10^{-7}sec. This corresponds to a propagating wave on the line of frequency about 10 MHz. This is usually unacceptable for the typical circuit employing silicon technology. Therefore, long lines usually lie on the metal level. For the case of large R or C of the line, multiple sections of a lumped circuit can be used to approximate the transmission line. In summary, for RC lossy lines, the criterion for the validity of the lumped circuit is that the RC time delay of the line be small relative to the period of the corresponding frequency of the line.

APPENDIX B

CAP USER'S MANUAL

B.1. Introduction

"CAP2D" and "CAP3D" are two FORTRAN programs for two-dimensional and three-dimensional capacitance computations. A detailed description of their usage and output format is given; some examples are also given at the end.

B.2. Description

The numerical computation of capacitance of these two programs is based on the integral method which is essentially the Green's function approach combined with the moment method. Pulse-type basis functions over subdivisions on the conductor's surface are assumed, and the testing functions are chosen the same as the basis functions. The subdivisions can be chosen to be variable-length or constant-length. If variable-length subdivisions are chosen, by default, they are divided according to the roots of the Chebyshev polynomials. Since subdivisions with variable length usually yield more accurate results for the same number of subdivisions, the constant case is basically included for comparison purposes.

Both programs have the capability to handle up to 10 traces (or conductors) in at most 3 layers of different media. In the two-dimensional case, "CAP2D" is also capable of taking into account the pitch angle of the slant side walls, assuming that the traces are left and right symmetric. However, "CAP3D" for the three-dimensional case deals with vertical side walls only. In the case of multiple layers, i.e., two layers or three layers of media, an infinite summation is involved in the Green's function and a cut-off criterion should be chosen.

B.2.1. Usage

The programs are designed to be run interactively. Alternatively, the UNIX system also provides a way to run it through an input file and redirect ("<") command. The following parameters will be prompted in running the programs.

(1) "number of traces" and "number of layers" : they are self-explanatory.

(2) "want kernel matrix ?" and "want grids ?" : If "y" is chosen, the kernel matrix elements and the coordinates of all the subdivisions will be listed in the output file. These are basically intended for debugging purposes. Usually "n" should be the answer to both requests.

(3) "wx,nx" : These are the horizontal or x-direction width of the trace and its corresponding number of subdivisions. Note that all the widths and lengths are stored in double precision variables, so "wx" should be in double precision form. This also applies to all the other widths in the program.

(4) "wy,ny","wz,nz" : These parameters are similar to the previous ones, except they are in vertical or y direction. "wz,nz" only appears in the three-dimensional case.

(5) "ang" : This is the pitch angle for the slant side walls. It only occurs in the two-dimensional case. The angle should be in degrees, and within the range of 0^o to 90^o.

(6) "subdiv method" : This concerns the choice of constant- or variable-length subdivisions. A "0" means constant subdivisions and any other integer will result in default Chebyshev variable subdivisions.

(7) "orientation" : This only occurs in running "CAP3D". It specifies the direction of the trace. For example, a trace which is parallel to the x-axis has an orientation 1. By the same token, the orientation is 2 if it is parallel to the y-axis. However, orientation 3 is prohibited. It should be pointed out that the charges on the cross sections perpendicular to the orientation axis of the trace are neglected, i.e., charges at each end of the line are not computed.

(8) "lolx,loly" : In "CAP2D", the coordinates of the lower left corner of the trace are needed.

(9) "cx,cy,cz" : On the other hand, the coordinates of the center of the trace are used in "CAP3D".

Note that the above parameters from (3) to (9) have to be furnished for every trace in question.

(10) "en1,en2" : These are for the dielectric constants of the media. Because the outmost layer in the three-layer case is assumed to be free space, at most two different dielectric constants need to be specified. It is important to note that the permittivity of the free space (8.854×10^{-14} F/cm) is not built into the programs. Thus, usually the output capacitance values are normalized by this factor. If capacitance values in Farad or F/cm, depending on three- or two-dimensional problem, are desired, then the permittivity of the free space should be included in the dielectric constants.

(11) "a,b,h" : These are the heights of the dielectric layers. They should be double precision numbers. "a" and "b" are for the case of three layers of media, and it is assumed that $b > a$. "H" is the height of the lower dielectric layer in the two-layer case.

(12) "errbnd" : This is the cut-off criterion for the infinite summation in the two-layer or three-layer Green's functions. 1.d-3 is recommended in the program. Yet, bear in mind that a small "errbnd" will render more computations, consume more CPU time, and result in more accurate answers. This trade-off between accuracy and computation is also observed in the case of the choice of the number of subdivisions for the traces.

B.2.2. Output

The result of the capacitance computation will be, by default, stacked in files "fort.7" for "CAP2D" and "fort.9" for "CAP3D". If these files already exist in the directory, the current output will be appended to the corresponding files. The report includes all the pertinent input parameters for the traces and the media. Furthermore, the following parameters are also included :

"Number of kernels" indicates the number of elements in the upper triangular part of the kernel matrix including diagonals. If the total number of subdivisions is n, then the "number of kernels" will be n*(n+1)/2. Note also that the kernel matrix will be listed in "packed symmetric" form if it is requested. In essence, the upper triangular and the diagonal elements are listed linearly. For example, the (i,j)th element ($i \leq j$) in the kernel matrix will be the [j*(j-1)/2+i]th element in the linear list.

"Iterm" indicates the maximum number of terms taken in the computation of multi-layer kernels. It is dependent on the cut-off criterion "errbnd" as discussed previously. And it is 0 for the single-layer case. Since the do loop index for the three-layer case is set at 12, it has not neccessarily converged if "iterm" equals 13. If divergence is encountered in the summation process, i.e., the absolute value of the term in the infinite summation is larger than its predecessor, "no. of divgnt" will be incremented, and the summation will be terminated. Since this divergence usually occurs at terms with small magnitude (≃1.d-2), the current sum can still be a reasonable value for the kernel element, and the computation of other kernel elements will continue.

The parameters "rcond" and "info" are indicators from the "linpack" programs "dppco" and "dppsl" which are utilized to solve the positive definite kernel matrix. If "info" is other than 0, the solution is erroneous. It indicates the order of the principal submatrix which is not positive definite. Usually the subdivision at fault can be detected this way. "Rcond" is the reciprocal of the condition number of the kernel matrix. Only when it is in the order of the machine epsilon (about 1.d-14 for double precision on VAX 11/780), then is the linear system ill-behaved and the solution may be erroneous. Usually this number is within 0.1 and 1.d-5.

The solution vector included in the report represents the even-mode charges on the subdivisions, i.e., the charge distribution when all the traces are at a constant positive potential with respect to the ground plane. It delineates the charge distribution in the system and is helpful in gaining a physical understanding of the system. Also it serves as a good gauge for debugging.

All the self capacitances, C_{nn}, and the coupling capacitances, C_{nm}, $n \neq m$ are reported. Note that the accuracy of the computed capacitances depends on the cutoff criterion "errbnd", the number of subdivisions and the subdivision method. Also it has been shown in the literature that the integral method will always result in a lower bound for the true capacitance value, and the true value is also variationally stationary for the first order perturbation. Thus, more subdivisions and a tighter "errbnd" will yield larger capacitance values. This characteristic is also very helpful for debugging, and for the estimation of the true value. Generally in order to ensure the accuracy of the capacitances, the same case can be run with successively more subdivisions on the dimensions of the traces.

Finally, the time usage shows the consumption of CPU time for various computations. "Kernel set up time" is the time to compute the kernel elements. It usually takes up the lion's share of the total user time. Since the "number of kernels" is the total number of elements computed, it is directly responsible for the "kernel setup time". "Kernel solve time" is the time spent by "dppco" and "dppsl" to solve the corresponding positive definite kernel matrix. "Capacitance time" is the time spent to obtain the various self and coupling capacitances from the solution vectors (i.e., charge distributions) of even-mode and odd-mode excitations. This time is usually the smallest and is negligible for single traces.

B.3. Files

The FORTRAN source files are : "cap3d.f", "subp3d0.f", "subp3d1.f", "subp3d2.f", "subp3d3.f" for the three-dimensional case. The main program is in "cap3d.f", all the others are subprograms. "subp3d1.f" contains the subprograms for the homogeneous case, "subp3d2.f" for the two-layer case and so on. The "make" file for "cap3d.ex" is as follows :

```
#
#       make file for cap3d.f
#
   FFLAGS = -g -u
   cap3d.ex : cap3d.o subp3d0.o subp3d1.o subp3d2.o subp3d3.o
              comptime.o
         f77 -g cap3d.o subp3d0.o subp3d1.o subp3d2.o subp3d3.o
              comptime.o -o cap3d.ex -llinpk
```

"Cap2d.ex" has a similar "make" file and "source" file structure, with names changed to 2d, accordingly. The object code "comptime.o" is the "object" file of the two C procedures which are utilized for the time usage calculation. "-llinpk" indicates that "linpack" is to be loaded, and this flag has to be the last flag in the command. Finally, the output will be in "fort.7" for "cap2d.ex" and "fort.9" for "cap3d.ex".

B.4. Warnings

The calculation of capacitances in three-layer cases is very computationally intensive, due to the infinite summation and the double summation inside. A few hundreds of seconds of CPU time on the UNIX system is not uncommon for a single trace (see example IV). So run this case cautiously and prudently.

B.5. Examples

In the following are some examples of interconnect systems. The configurations of interconnects are presented in graphs, and the interactive sessions of running the programs and the results from the run are also included. Hopefully, these examples will illustrate the usage of both programs in more detail.

B.5.1. Example 1

Following is the interactive session of running the program "cap3d" for the traces in Fig. B.1.

```
%
% cap3d.ex
 number of traces       ntrc = 2
 number of layers       nlay = 1
 want kernel matrix ('y' or 'n') ? 'n'
 want grids ('y' or 'n') ? 'n'
 trace no. 1 :
 orientation (x=1,y=2) = 1
 subdivision (const=0, Cheby=other) = 1
 x width and no. of div.  :  wx,nx = 6.5d0 4
 y width and no. of div.  :  wy,ny = 1.3d0 3
 z width and no. of div.  :  wz,nz = 1.2d0 3
 position of center : cx,cy,cz = 0.d0 0.d0 1.1d0
 trace no. 2 :
 orientation (x=1,y=2) = 2
 subdivision (const=0, Cheby=other) = 1
 x width and no. of div.  :  wx,nx = 1.8d0 3
 y width and no. of div.  :  wy,ny = 9.d0 4
 z width and no. of div.  :  wz,nz = 1.7d0 3
 position of center : cx,cy,cz = 0.d0 0.d0 3.35d0
 dielectric const. : enl = 4.d0
STOP in fort.9 statement executed
```

END of the interactive session.

The output from the above session is as follows :

```
***** three-dimensional,  1-layer,  2-trace ***** Apr 1 83  00:17:29
    subdivisions : (x, y and z)
         .44   2.81   2.81    .44    .19    .92    .19
         .18    .85    .18

    trace no. 1 with wx = 6.50  nx =  4     wy = 1.30  ny =  3
     wz = 1.20  nz =  3     center at :   .000   .000  1.100
      orientation : 1   subdiv. method :  1

    subdivisions : (x, y and z)
         .26   1.27    .26    .60   3.90   3.90    .60
```

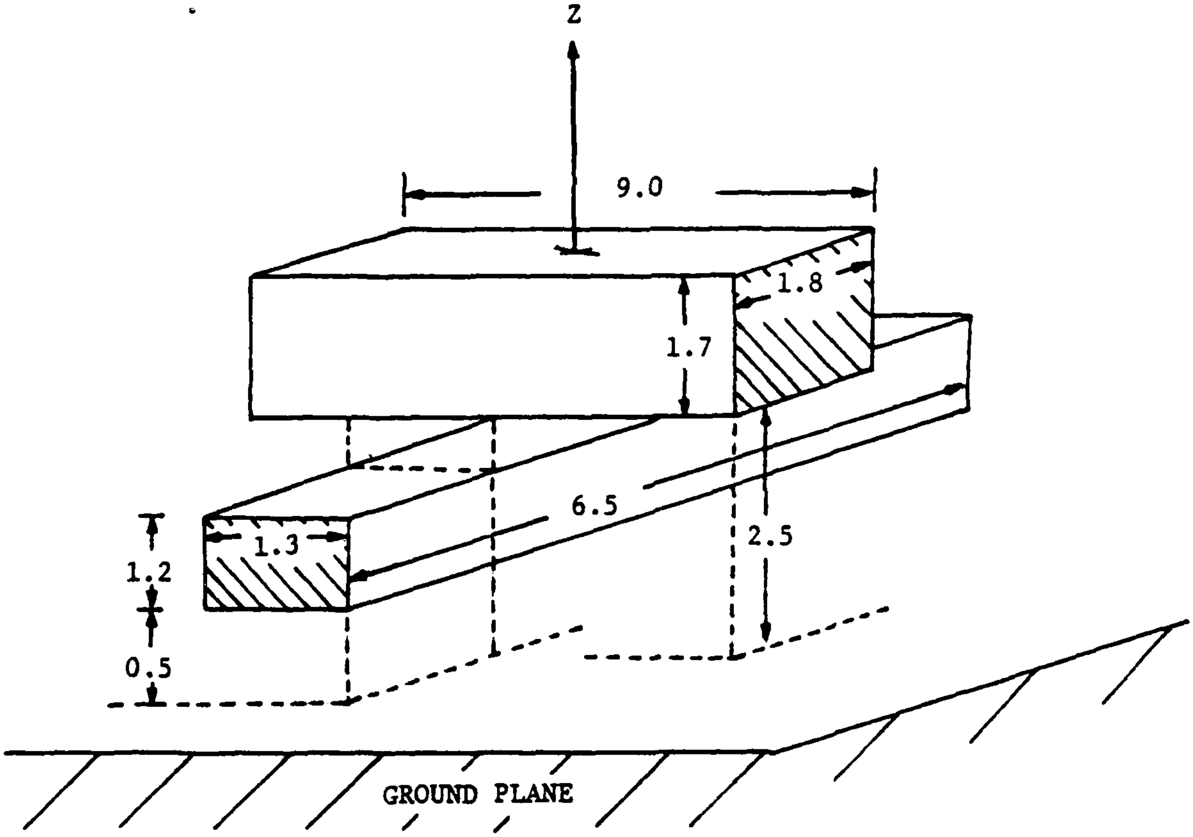


Figure B.1 Three-dimensional, cross-over configuration of two traces.

```
      .25   1.20    .25

   trace no. 2 with wx = 1.80  nx =  3     wy = 9.00  ny =  4
    wz = 1.70  nz =  3     center at :   .000   .000  3.350
     orientation : 2   subdiv. method :  1

   1-layer with diel. const. enl =   4.00

   total no. of subsections :    96   no. of kernels :  4656
    iterm =     0  no. of divgnt :     0
   areas :
   .08    .54    .54    .08    .40   2.59   2.59    .40    .08    .54
   .54    .08    .08    .54    .54    .08    .40   2.59   2.59    .40
   .08    .54    .54    .08    .08    .49    .49    .08    .37   2.39
  2.39    .37    .08    .49    .49    .08    .08    .49    .49    .08
   .37   2.39   2.39    .37    .08    .49    .49    .08    .16    .77
   .16   1.03   4.96   1.03   1.03   4.96   1.03    .16    .77    .16
   .16    .77    .16   1.03   4.96   1.03   1.03   4.96   1.03    .16
   .77    .16    .15    .97    .97    .15    .72   4.68   4.68    .72
   .15    .97    .97    .15    .15    .97    .97    .15    .72   4.68
  4.68    .72    .15    .97    .97    .15


    rcond =     .21049453e-01   info =     0
    solution(even) :
  1.447        6.351        6.351        1.447        5.809        20.83
  20.83        5.809        1.447        6.351        6.351        1.447
  .7007        .6746        .6746        .7007        2.876        .6321
  .6321        2.876        .7007        .6746        .6746        .7007
  1.210        4.889        4.889        1.210        3.374        6.229
  6.229        3.374        .6668        .8890        .8890        .6668
  1.210        4.889        4.889        1.210        3.374        6.229
  6.229        3.374        .6668        .8890        .8890        .6668
  1.277        5.118        1.277        2.299        5.510        2.299
  2.299        5.510        2.299        1.277        5.118        1.277
  1.016        4.028        1.016        2.639        6.598        2.639
  2.639        6.598        2.639        1.016        4.028        1.016
  1.196        2.281        2.281        1.196        4.189        6.039
  6.039        4.189        .9724        2.594        2.594        .9724
  1.196        2.281        2.281        1.196        4.189        6.039
  6.039        4.189        .9724        2.594        2.594        .9724

    no. of traces :      2
    cap( 1, 1) =    166.0239059      cap( 2, 2) =    140.5242337
    cap( 1, 2) =    51.89136663

TIME USAGE : kernel setup  time : 346.467
             kernel solve  time : 5.93333
             capacitance   time : .283333
             total user    time : 352.683
```

```
**********   end of report   **********   Apr 1 83   00:27:43
```

B.5.2. Example 2

The interactive session and output of running "cap2d" for the traces in Fig. B.2 :

```
%
% cap2d.ex
 number of traces        ntrc = 4
 number of layers        nlay = 2
 want kernel matrix ('y' or 'n') ? 'n'
 want grids ('y' or 'n') ? 'n'
 trace no.  1 :
 horiz. width and no. of div.  :  wx,nx = 1.5d0 3
 verti. width and no. of div.  :  wy,ny = .44d0 3
 position of low left corner: lolx,loly = 2.d0 .38d0
 slant angle of the trace : ang (in deg) = 7.0d1
 division method (const=0 Cheby=other) = 1
 trace no. 2 :
 horiz. width and no. of div.  :  wx,nx = 1.5d0 3
 verti. width and no. of div.  :  wy,ny = .44d0 3
 position of low left corner: lolx,loly = 4.5d0 .38d0
 slant angle of the trace : ang (in deg) = 7.d1
 division method (const=0 Cheby=other) = 1
 trace no. 3 :
 horiz. widch and no. of div.  :  wx,nx = 2.2d0 3
 verti. width and no. of div.  :  wy,ny = 1.7d0 3
 position of low left corner: lolx,loly = 0.d0 2.37d0
 slant angle of the trace : ang (in deg) = 7.5d1
 division method (const=0 Cheby=other) = 1
 trace no. 4 :
 horiz. width and no. of div.  :  wx,nx = 2.2d0 3
 verti. width and no. of div.  :  wy,ny = 1.7d0 3
 position of low left corner: lolx,loly = 3.2d0 2.37d0
 slant angle of the trace : ang (in deg) = 7.5d1
 division method (const=0 Cheby=other) = 1
 diel. const. of lower layer : en1 = 4.d0
 diel. const. of higher layer: en2 = 7.5d0
 interface position : h = 1.07d0
 relative error bound : errbnd (1.d-3) = 1.d-3
STOP in tort.7 statement executed
```

END of the interactive session.

```
******* two-dimensional, 2-layer, 4-trace *******  Apr15 83   01:56:11

     trace no. 1 with wx = 1.50  nx =  3        wy =  .44   ny =  3
     position :  2.00    .38          slant angle (deg) :   70.00
     subsections :            .173   .834   .173   .220  1.061   .220
```

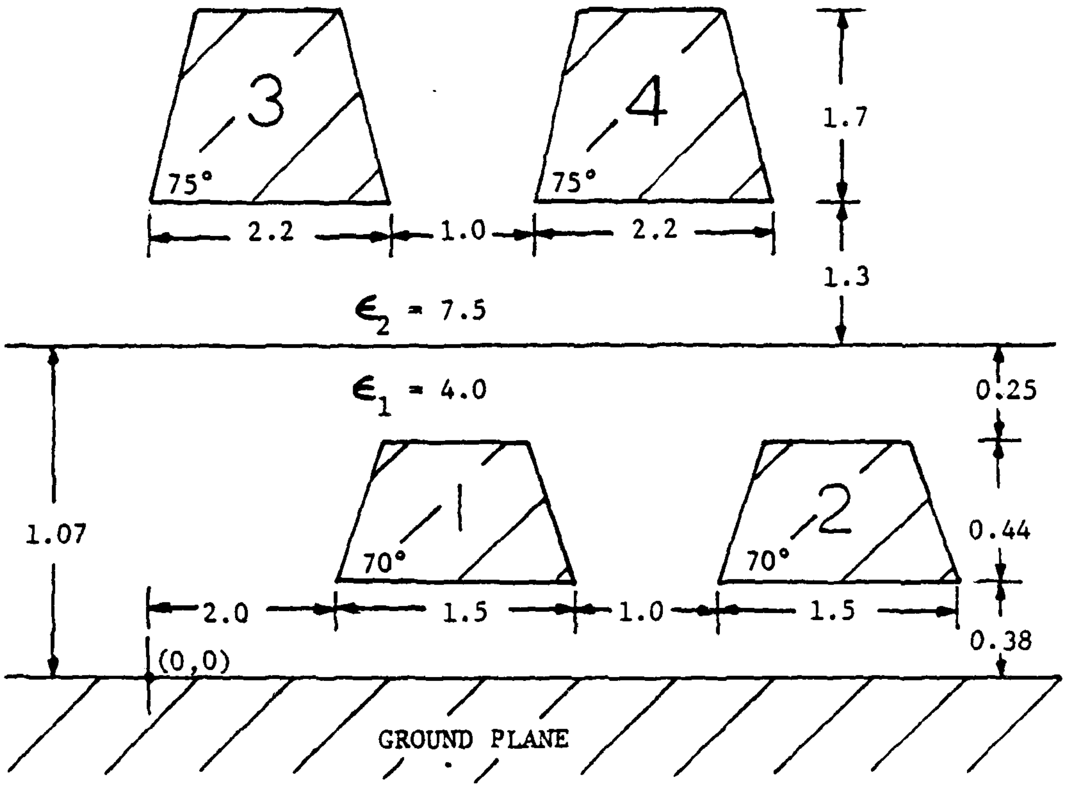


Figure B.2 Four two-dimensional traces in two layers of media.

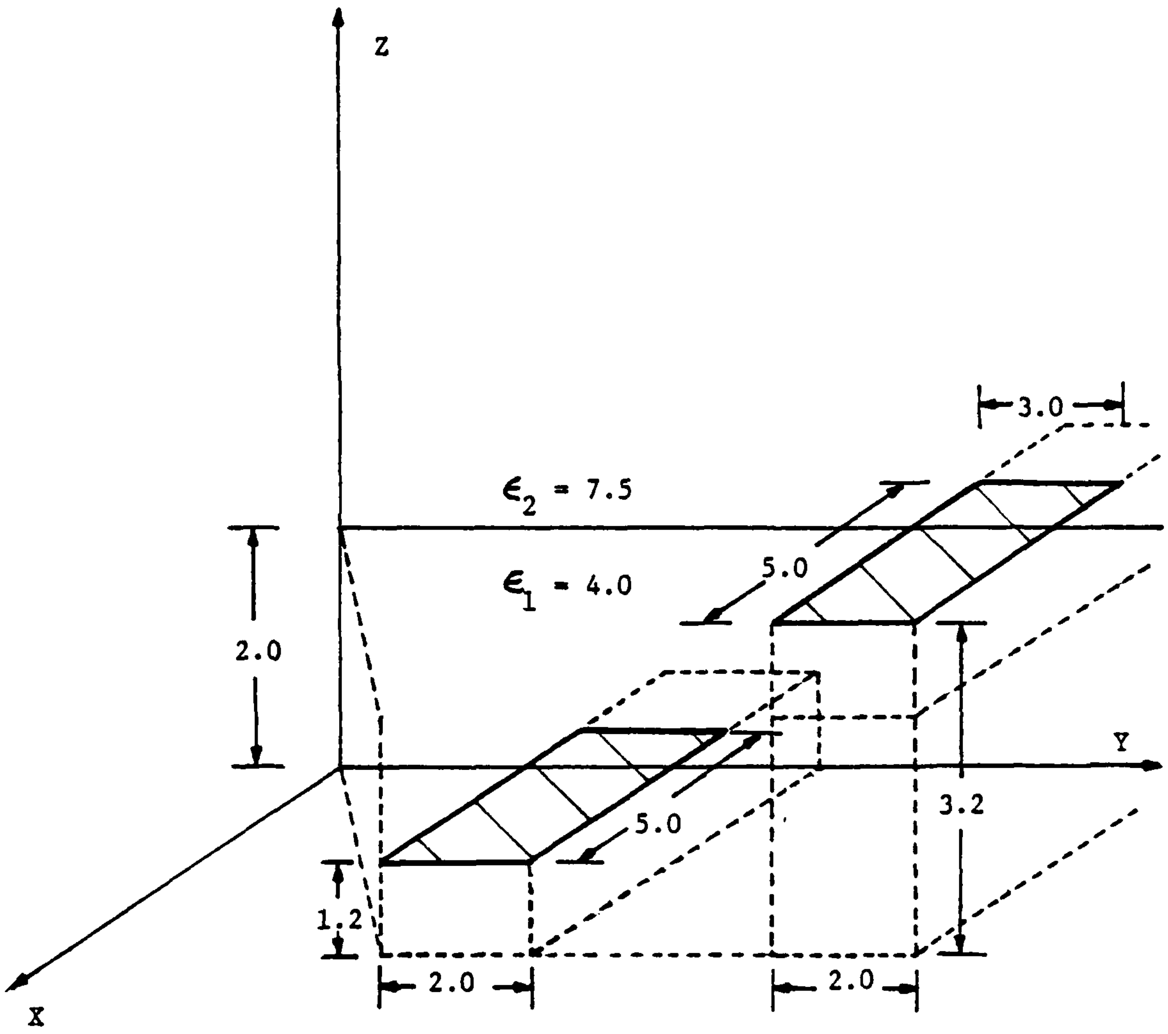
```
      .069    .331    .069

    trace no. 2 with wx = 1.50  nx =  3        wy =  .44   ny =  3
    position :  4.50    .38            slant angle (deg) :   70.00
    subsections :               .173   .834    .173   ,220  1.061   .220
      .069    .331    .069

    trace no. 3 with wx = 2.20  nx =  3        wy = 1.70   ny =  3
    position :   .00  2.37             slant angle (deg) :   75.00
    subsections :               .189   .911    .189   .322  1.556   .322
      .258  1.244    .258

    trace no. 4 with wx = 2.20  nx =  3        wy = 1.70   ny =  3
    position :  3.20  2.37             slant angle (deg) :   75.00
    subsections :               .189   .911    .189   .322  1.556   .322
      .258  1.244    .258

    2-layer with brdy h =    1.07      rel errbnd =  .100e-02
            diel. const. enl(low) =   4.00    en2(top) =   7.50

    total no. of subsections :    48   no. of kernels :  1176
    iterm =     7   no. of divgnt :    0

    widths :
   .22   1.06    .22    .17    .83    .17    .07    .33    .07    .07
   .33    .07    .22   1.06    .22    .17    .83    .17    .07    .33
   .07    .07    .33    .07    .32   1.56    .32    .19    .91    .19
   .26   1.24    .26    .26   1.24    .26    .32   1.56    .32    .19
   .91    .19    .26   1.24    .26    .26   1.24    .26

   rcond =     .24604396e-01   info =     0
   solution(even) :
   2.982        11.14        2.911        .2795        .3718        .1257
   1.387        1.317        .2139        1.248        .9462        .1012
   2.910        11.14        3.008        .1394        .7297        .5101
   1.249        .9512        .1137        1.441        1.565        .3493
   1.759        2.804        .1911        .6801        1.291        .3642
   1.494        2.748        .7878        .9962e-01    .2684        .3656
   .1260        .7004        .6998        .3601        1.254        .6398
   .7113e-01    .3019        .3588        .8920        2.231        .7506

   no. of traces :     4
   cap( 1, 1) =   23.02923288       cap( 2, 2) =   24.11008633
   cap( 3, 3) =   12.85232106       cap( 4, 4) =   [illegible].386332101
   cap( 1, 2) =   1.766451058       cap( 1, 3) =   4.719475070
   cap( 1, 4) =   4.522507157       cap( 2, 3) =   .2622865412
   cap( 2, 4) =   7.549320560       cap( 3, 4) =   15.43729017

TIME USAGE : kernel setup  time : 198.850
             kernel solve  time : 1.15000
             capacitance   time : .233333
```

```
              total user    time : 200.233

**********    end of report    **********   Apr15 83   02:03:17
```

B.5.3. Example 3

The interactive session and output of running "cap3d" for the traces in Fig. B.3 :

```
%
% cap3d.ex
 number of traces       ntrc = 2
 number of layers       nlay = 2
 want kernel matrix ('y' or 'n') ? 'n'
 want grids ('y' or 'n') ? 'n'
 trace no. 1 :
 orientation (x=1,y=2) = 1
 subdivision (const=0, Cheby=other) = 1
 x width and no. of div.  :  wx,nx = 5.d0 3
 y width and no. of div.  :  wy,ny = 2.d0 3
 z width and no. of div.  :  wz,nz = 0.d0 0
 position of center : cx,cy,cz = 3.1d0 3.d0 1.2d0
 trace no. 2 :
 orientation (x=1,y=2) = 1
 subdivision (const=0, Cheby=other) = 1
 x width and no. of div.  :  wx,nx = 5.d0 3
 y width and no. of div.  :  wy,ny = 3.d0 3
 z width and no. of div.  :  wz,nz = 0.d0 0
 position of center : cx,cy,cz = 3.1d0 6.5d0 3.2d0
 diel. const. of lower layer : en1 = 4.d0
 diel. const. of higher layer: en2 = 7.5d0
 interface position : h = 2.d0
 relative error bound : errbnd (1.d-3) = 1.d-3
STOP in fort.9 statement executed

END of interactive session.


******* three-dimensional, 2-layer, 2-trace ******* Apr15 83 01:38:41

     subdivisions : (x, y and z)
         .73   3.54    .73    .29   1.41    .29

     trace no. 1 with wx = 5.00  nx =  3     wy = 2.00  ny =  3
     wz =  .00  nz =  0     center at :  3.100  3.000  1.200
     orientation : 1   subdiv. method :  1

     subdivisions : (x, y and z)
         .73   3.54    .73    .44   2.12    .44
```



GROUND PLANE

Figure B.3 Two thin plates in two layers of media.

```
    trace no. 2 with wx = 5.00  nx =  3     wy = 3.00  ny =  3
    wz =  .00  nz =  0     center at :  3.100  6.500  3.200
    orientation : 1   subdiv. method :  1

    2-layer with brdy h =   2.00     rel errbnd =  .100e-02
               diel. const. enl(low) =   4.00   en2(top) =   7.50

    total no. of subsections :    18   no. of kernels :   171
    iterm =     6   no. of divgnt :     0

    areas :
  .21   1.04    .21   1.04   5.00   1.04    .21   1.04    .21    .32
 1.55    .32   1.55   7.50   1.55    .32   1.55    .32


    rcond =     .53305083e-01   info =     0
    solution(even) :
    4.092        14.04        4.092        9.938        27.23        9.938
    3.553        11.40        3.553        6.053        16.18        6.053
    17.37        35.94        17.37        7.045        20.92        7.045

    no. of traces :     2
    cap( 1, 1) =    87.83536715      cap( 2, 2) =   133.9861517
    cap( 1, 2) =    15.45701657


TIME USAGE : kernel setup  time : 65.4833
             kernel solve  time : .183333
             capacitance   time : .166667e-01
             total user    time : 65.6833

**********   end of report   **********   Apr15 83    01:42:41
```

B.5.4. Example 4

The interactive session and the output for the trace in Fig. B.4 :

```
%
% cap2d.ex
 number of traces       ntrc = 1
 number of layers       nlay = 3
 want kernel matrix ('y' or 'n') ? 'y'
 want grids ('y' or 'n') ? 'n'
 trace no. 1 :
 horiz. width and no. of div.  :  wx,nx = 1.7d0 3
 verti. width and no. of div.  :  wy,ny = 1.2d0 3
 position of low left corner: lolx,loly = 0.d0 .5d0
 slant angle of the trace : ang (in deg) = 7.5d1
```

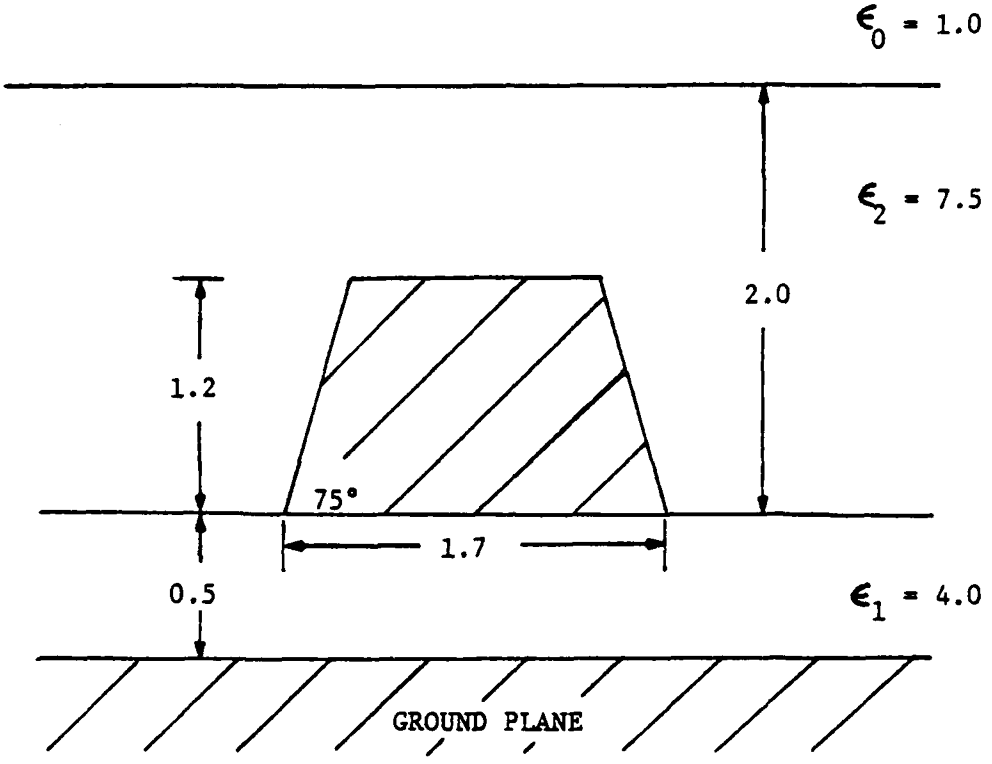


Figure B.4 A two-dimensional trace in three layers of media.

```
 division method (const=0 Cheby=other) = 1
 diel. const. of lower layer : en1 = 4.d0
 diel. const. of higher layer: en2 = 7.5d0
 interface positions : a,b (a<b) = 0.5d0 2.5d0
 relative error bound : errbnd (1.d-3) = 1.d-2
STOP in fort.7 statement executed

END of interactive session.


******* two-dimensional, 3-layer, 1-trace ******* Apr15 83   00:02:56

     trace no. 1 with wx = 1.70  nx =  3       wy = 1.20   ny =  3
     position :   .00   .50          slant angle (deg) :   75.00
     subsections :            .155   .747   .155   .249  1.202   .249
        .182   .878   .182

     3-layer with bdry heights a =    .50      b =  2.50
                  diel const en1(low) =   4.00   en2(top) =   7.50
                  relative errbnd =  .100e-01

     total no. of subsections :    12   no. of kernels :    78
     iterm =    12   no. of divgnt :     0

     widths :
    .25   1.20    .25    .15    .75    .15    .18    .88    .18    .18
    .88    .18

     packed symm matrix ker :
   .84743e-01  .20391e-01  .40871e-01  .37890e-02  .20391e-01  .84743e-01
   .18020e-01  .16603e-01  .11365e-01  .99889e-01  .15007e-01  .17498e-01
   .15007e-01  .47521e-01  .65681e-01  .11365e-01  .16603e-01  .18020e-01
   .27945e-01  .47521e-01  .99889e-01  .62266e-01  .17036e-01  .37810e-02
   .19381e-01  .15633e-01  .11525e-01  .88393e-01  .30659e-01  .18603e-01
   .81513e-02  .35820e-01  .26691e-01  .18518e-01  .35191e-01  .55031e-01
   .19202e-01  .16595e-01  .10561e-01  .70566e-01  .39781e-01  .24433e-01
   .20815e-01  .40914e-01  .95082e-01  .37810e-02  .17036e-01  .62266e-01
   .11525e-01  .15633e-01  .19381e-01  .33764e-02  .77381e-02  .10605e-01
   .88393e-01  .81513e-02  .18603e-01  .30659e-01  .18518e-01  .26691e-01
   .35820e-01  .77381e-02  .12357e-01  .16755e-01  .35191e-01  .55031e-01
   .10561e-01  .16595e-01  .19202e-01  .25121e-01  .40466e-01  .68090e-01
   .10605e-01  .16755e-01  .22312e-01  .20815e-01  .40914e-01  .95082e-01

     rcond =      .61052279e-01    info =      0
     solution(even) :
     2.958        11.40        2.958        .9070        2.174        .9881
     3.017        5.438        1.200        3.017        5.430        1.156

     no. of traces :     1
     cap( 1, 1) =   40.64073429
```

```
TIME USAGE : kernel setup  time : 970.250
             kernel solve  time : .133333
             capacitance   time : .000000e+00
             total user    time : 970.383

***********   end of report   ***********   Apr15 83   00:28:16
```

APPENDIX C

COMPUTATION OF TWO-DIMENSIONAL KERNEL MATRIX ELEMENTS

C.1. Introduction

In reference to Eq. (2.14) the kernel matrix element A_{ij} is a double integral containing the Green's function, the basis function and the testing function. In order to have a closed form for the evaluation of these elements, the basis functions are chosen as pulse-type functions, and the testing functions are the same as the basis functions for Galerkin's method. Two types of interconnect side walls are considered in the two-dimensional case, i.e., vertical and slant side walls. Of course, the vertical case can be considered as a special case of the slant side wall with a pitch of 90^o. For the sake of comparison, both cases are considered. Additionally, let us define a special convention to simplify the expressions of those closed forms.

$$\Big[f(x,y)\Big]_{x=\alpha_1,\ \alpha_3}^{\alpha_2,\ \alpha_4} = f(\alpha_2,y) - f(\alpha_1,y) + f(\alpha_4,y) - f(\alpha_3,y) \qquad (C.1)$$

$$\Big[f(x,y)\Big]_{x=\alpha_1,\ y=\beta_1}^{\alpha_2 \qquad \beta_2} = f(\alpha_2,\beta_2) - f(\alpha_2,\beta_1) - f(\alpha_1,\beta_2) + f(\alpha_1,\beta_1) \qquad (C.2)$$

where f(x,y) can be any real-valued function, and α_1, α_2, α_3, α_4, β_1, β_2 are real constants.

C.2. Vertical Side Wall

For the two-dimensional and vertical side wall case, the basis functions which divide the perimeter of the cross-section of an interconnect into subdivisions can generally be categorized into two groups : horizontal pulses and vertical pulses. They are defined as

$$P_j^h(x,y) = \begin{cases} 1/S_j, & \text{for } x_j^- \le x \le x_j^+,\ y = y_j^o \\ 0, & \text{otherwise} \end{cases} \qquad \text{(C.3)}$$

$$\text{and } S_j = (x_j^+ - x_j^-)^{*}$$

$$P_j^v(x,y) = \begin{cases} 1/S_j, & \text{for } y_j^- \le y \le y_j^+,\ x = x_j^o \\ 0, & \text{otherwise} \end{cases} \qquad \text{(C.4)}$$

$$\text{and } S_j = (y_j^+ - y_j^-)^{*}$$

Since the "cores" of the two-dimensional Green's functions in Eqs. (2.15,2.16,2.21,2.22) are logarithmic, the following integrals result from incorporating the pulses in Eq. (C.3) and Eq. (C.4). Assume that $x_j^+ \ge x_j^-$, $x_i^+ \ge x_i^-$, $y_j^+ \ge y_j^-$, $y_i^+ \ge y_i^-$ and A, K are real constants which can also be zeros; moreover, x_i^o is the abscissa of the vertical pulse P_i^v, and y_j^o is the ordinate of the horizontal pulse P_j^h.

$$\text{(1) } I_{xx} = \int_{x_j^-}^{x_j^+} dx \int_{x_i^-}^{x_i^+} dx' \ \ln[(x-x')^2 + A^2]$$

* Normalizing the pulse amplitude with the pulse width or area, S_j, is essentially the same as using the charge Q as the "state variable" as described in [10]. The advantage is a numerically more stable kernel matrix.

$$(2)\quad I_{yy1} = \int_{y_j^-}^{y_j^+} dy \int_{y_i^-}^{y_i^+} dy' \;\ln[A^2 + (y+y'+K)^2]$$

$$(3)\quad I_{yy2} = \int_{y_j^-}^{y_j^+} dy \int_{y_i^-}^{y_i^+} dy' \;\ln[A^2 + (y-y'+K)^2]$$

$$(4)\quad I_{yy3} = \int_{y_j^-}^{y_j^+} dy \int_{y_i^-}^{y_i^+} dy' \;\ln[A^2 + (|y-y'|+K)^2]$$

$$(5)\quad I_{xy1} = \int_{x_j^-}^{x_j^+} dx \int_{y_i^-}^{y_i^+} dy' \;\ln[(x-x_i^0)^2 + (y_j^0+y'+K)^2]$$

$$(6)\quad I_{xy2} = \int_{x_j^-}^{x_j^+} dx \int_{y_i^-}^{y_i^+} dy' \;\ln[(x-x_i^0)^2 + (y_j^0-y'+K)^2]$$

$$(7)\quad I_{xy3} = \int_{x_j^-}^{x_j^+} dx \int_{y_i^-}^{y_i^+} dy' \;\ln[(x-x_i^0)^2 + (|y_j^0-y'|+K)^2]$$

Because of the similarity of the above integrals, the closed-form expressions for them can be given in terms of the following functions :

$$F(x,y) = \frac{1}{2}(x^2-y^2)\ln(x^2+y^2) - \frac{3}{2}\,x^2 + 2xy\tan^{-1}\frac{x}{y} \qquad \text{(C.5)}$$

$$H(x,y) = xy\ln(x^2+y^2) - 4xy + 2y^2\tan^{-1}\frac{x}{y} \qquad \text{(C.6)}$$

$$G(\alpha,\beta,a) = \int_{\alpha}^{\beta} x\tan^{-1}\frac{a}{x}dx, \qquad \beta \geq \alpha$$

$$= \begin{array}{ll} 0.5[(a^2+x^2)\tan^{-1}\frac{a}{x} + ax]\Big|_{x=\alpha}^{\beta}, & \beta>\alpha\geq 0 \text{ or } 0\geq\beta>\alpha \\ 0.5[(a^2+x^2)\tan^{-1}\frac{a}{x} + ax]\Big|_{x=\alpha}^{\beta} - \frac{\pi}{2}a^2\,\mathrm{sign}(a), & \alpha<0\leq\beta \end{array} \qquad \text{(C.7)}$$

where $\mathrm{sign}(a) = \dfrac{a}{|a|}$.

Note that F(x,y) is an even function in both variables x,y, while H(x,y) is an odd function on both variables. The special consideration given to G(α,β,a) as described above is due to the singularity or branch cut at x=0.

After some algebraic manipulation, the integrals, I_{xx}, I_{yy1}, etc., can be expressed in terms of Eqs. (C.5,C.6,C.7) as follows :

$$(1)\quad I_{xx} = -\Big[F(x,A)\Big]_{x=x_j^+-x_i^-,\ x_j^--x_i^+}^{x_j^+-x_i^+,\ x_j^--x_i^-}$$

$$(2)\quad I_{yy1} = \Big[F(y,A)\Big]_{y=y_j^++y_i^-+K,\ y_j^-+y_i^++K}^{y_j^++y_i^++K,\ y_j^-+y_i^-+K}$$

$$(3)\quad I_{yy2} = -\Big[F(y,A)\Big]_{y=y_j^+-y_i^-+K,\ y_j^--y_i^++K}^{y_j^+-y_i^++K,\ y_j^--y_i^-+K}$$

$$(\mathrm{i})\ -\Big[F(y,A)\Big]_{y=|y_j^+-y_i^-|+K,\ |y_j^--y_i^+|+K}^{|y_j^+-y_i^+|+K,\ |y_j^--y_i^-|+K}$$

$$y_j^+\geq y_j^-\geq y_i^+\geq y_i^- \text{ or } y_i^+\geq y_i^-\geq y_j^+\geq y_j^-$$

$$(\mathrm{ii})\ \Big[F(y,A)\Big]_{y=y_j^--y_i^-+K,\ y_i^+-y_j^++K}^{y_j^+-y_i^-+K,\ y_i^+-y_j^-+K} - 2(y_j^+-y_j^-)\cdot$$

$$(K\ln(K^2+A^2)-2K+2A\tan^{-1}\frac{K}{A}),\quad y_i^+\geq y_j^+\geq y_j^-\geq y_i^-$$

(4) I_{yy3} = (iii) interchange $y_j^- \langle\!-\!\rangle\, y_i^-$, $y_j^+ \langle\!-\!\rangle\, y_i^+$

in the above expression, if $y_j^+\geq y_i^+\geq y_i^-\geq y_j^-$

$$\text{(iv)}\quad \Big[F(y,A)\Big]_{y=y_i^- - y_j^- + K,\ y_i^+ - y_j^+ + K}^{y_j^+ - y_i^- + K,\ y_i^+ - y_j^- + K} - 2(y_j^+ - y_i^-)\cdot$$

$$(K\ln(K^2+A^2)-2K+2A\tan^{-1}\frac{K}{A}),\quad y_i^+ > y_j^+ > y_i^- > y_j^-$$

(v) interchange $y_j^- <-> y_i^-$, $y_j^+ <-> y_i^+$

in the above expression, if $y_j^+ > y_i^+ > y_j^- > y_i^-$

$$\text{(5)}\quad I_{xy1} = \Big[[H(x,y)]_{x=x_j^- - x_i^o}^{x_j^+ - x_i^o} + 2G(x_j^+ - x_i^o, x_j^- - x_i^o, y)\Big]_{y=y_j^o + y_i^- + K}^{y_j^o + y_i^+ + K}$$

$$\text{(6)}\quad I_{xy2} = -\Big[[H(x,y)]_{x=x_j^- - x_i^o}^{x_j^+ - x_i^o} + 2G(x_j^+ - x_i^o, x_j^- - x_i^o, y)\Big]_{y=y_j^o - y_i^- + K}^{y_j^o - y_i^+ + K}$$

$$\text{(7)}\quad I_{xy3} = \begin{cases} \operatorname{sign}(y_i^- - y_j^o)\Big[[H(x,y)]_{x=x_j^- - x_i^o}^{x_j^+ - x_i^o} \\ \qquad + 2G(x_j^+ - x_i^o, x_j^- - x_i^o, y)\Big]_{y=|y_j^o - y_i^-|+K}^{|y_j^o - y_i^+|+K} \\ \qquad y_j^o \geq y_i^+ \text{ or } y_j^o \leq y_i^- \\ \Big[[H(x,y)]_{x=x_j^- - x_i^o}^{x_j^+ - x_i^o} + 2G(x_j^+ - x_i^o, x_j^- - x_i^o, y)\Big]_{y=\ K,\quad K}^{y_j^o - y_i^- + K,\ y_i^+ - y_j^o + K} \\ \qquad y_i^+ > y_j^o > y_i^- \end{cases}$$

C.3. Slant Side Wall

If the pitch angles of the side walls of the interconnects are taken into account, assuming that the interconnects are left and right symmetric, the pulse-type basis functions, be they on the horizontal planes or on the slant side walls, generally can be expressed parametrically with the help of polar coordinates as

$$P_j^v(x,y) = \begin{cases} 1/\Delta_j, & \text{for } x=x_j+t\cos\eta_j,\ y=y_j+t\sin\eta_j,\ 0 \le t \le \Delta_j \\ 0, & \text{otherwise} \end{cases} \tag{C.7}$$

Note that (x_j,y_j) is the coordinate of one of the end points of the subdivision, and η_j is the polar angle of the subdivision. This angle should be limited within 0° and 360°. Δ_j is the width of this subdivision, and as in the previous case, it is used to normalize the pulse amplitude. For horizontal subdivisions, the angle $\eta_j = 0^\circ$ or 180°. Hence, all the subdivisions on the two-dimensional cross-section of an interconnect can be represented by Eq. (C.7).

As in the previous section, after incorporating the "cores" of the two-dimensional Green's functions with the pulse in Eq. (C.7), the following integrals are needed for the evaluation of the kernel matrix elements.

(1) $$I_1 = \int_0^{\Delta_j} dt \int_0^{\Delta_i} dt' \ \ln[(x_j+t\cos\eta_j-x_i-t'\cos\eta_i)^2 + (y_j+t\sin\eta_j-y_i-t'\sin\eta_i+k_0)^2]$$

(2) $$I_2 = \int_0^{\Delta_j} dt \int_0^{\Delta_i} dt' \ \ln[(x_j+t\cos\eta_j-x_i-t'\cos\eta_i)^2 + (y_j+t\sin\eta_j+y_i+t'\sin\eta_i+k_0)^2]$$

(3) $$I_3 = \int_0^{\Delta_j} dt \int_0^{\Delta_i} dt' \ \ln[(x_j+t\cos\eta_j-x_i-t'\cos\eta_i)^2 + (|y_j+t\sin\eta_j-y_i-t'\sin\eta_i|+k_0)^2]$$

where k_0 is a real constant, and Δ_j, Δ_i are the lengths of the jth and the ith subdivisions in question.

In order to encompass the evaluation of all three of the above integrals, let us first consider a more general integral, assuming α, a, b and k_0 are all real constants.

$$I = \int_0^{\Delta_j} dt \int_\alpha^{a+bt} dt' \; \ln[(x_j+t\cos\eta_j-x_i-t'\cos\eta_i)^2 + (y_j+t\sin\eta_j-y_i-t'\sin\eta_i+k_0)^2]$$

$$= \int_0^{\Delta_j} dt \int_{\alpha-k_2}^{a+bt-k_2} dt' \; \ln(t'^2+k_1^2) \qquad (C.8a)$$

$$\text{where } k_1 = \alpha_s t + \gamma_1$$
$$k_2 = \alpha_c t + \gamma_2 \qquad (C.8b)$$

$$\text{and } \alpha_s = -\sin(\eta_j-\eta_i) \qquad \alpha_c = \cos(\eta_j-\eta_i)$$
$$\gamma_1 = (x_j-x_i)\sin\eta_i - (y_j-y_i+k_0)\cos\eta_i$$
$$\gamma_2 = (x_j-x_i)\cos\eta_i + (y_j-y_i+k_0)\sin\eta_i \qquad (C.8c)$$

Furthermore, two more functions are defined to represent the integral in Eq. (C.8a). They are :

$$F_1(\Delta_j,a,b) = \int_0^{\Delta_j} dt \; (\alpha_s t+\gamma_1)\tan^{-1}\frac{(b-\alpha_c)t+(a-\gamma_2)}{\alpha_s t+\gamma_1} \qquad (C.9a)$$

$$F_2(\Delta_j,a,b) = \int_0^{\Delta_j} dt\ [(b-\alpha_c t+(a-\gamma_2))]\ \ln\{[(b-\alpha_c)t+(a-\gamma_2)]^2+(\alpha_s t+\gamma_1)^2\} \tag{C.9b}$$

Finally, the integral I in Eq. (C.8a) can be written in terms of the above F_1 and F_2 as

$$\begin{aligned} I = {} & 2F_1(\Delta_j,a,b) + F_2(\Delta_j,a,b) - 2F_1(\Delta_j,\alpha,0) - F_2(\Delta_j,\alpha,0) \\ & - 2(a-\alpha)\Delta_j - b\Delta_j^2 \end{aligned} \tag{C.10}$$

After some lengthy derivation, the integrals in Eq. (C.9) can be expressed in closed forms as :

$$F_1(\Delta_j,a,b) = \begin{cases} \text{(i)}\ (\frac{\alpha_s}{2}\Delta_j^2 + \gamma_1\Delta_j)\tan^{-1}\frac{(b-\alpha_c)\Delta_j+(a-\gamma_2)}{\alpha_s\Delta_j+\gamma_1}, \quad \text{if } \gamma_6 = 0 \\[2ex] \text{(ii)}\ (\frac{\alpha_s}{2}\Delta_j^2 + \gamma_1\Delta_j)\tan^{-1}\frac{(b-\alpha_c)\Delta_j+(a-\gamma_2)}{\alpha_s\Delta_j+\gamma_1} - \frac{\alpha_s\gamma_6}{2\gamma_7}\Delta_j \\ \quad - \frac{\gamma_6^2}{2\gamma_7^2}(b-\alpha_c)\ \ln\left| \frac{\gamma_7\Delta_j^2+2\gamma_8\Delta_j+\gamma_9}{\gamma_9} \right| \\ \quad + \frac{1}{\gamma_7}[\frac{\alpha_s\gamma_9}{2} + \frac{\gamma_6\gamma_8(b-\alpha_c)}{\gamma_7}]. \\ \quad (\tan^{-1}\frac{\gamma_8+\gamma_7\Delta_j}{\gamma_6} - \tan^{-1}\frac{\gamma_8}{\gamma_6}), \text{ if } \gamma_6 \neq 0 \end{cases} \tag{C.11a}$$

$$\text{(i)}\ (a-\gamma_2)\Delta_j\ \ln|\gamma_9|, \quad \text{if } \gamma_7 = 0 \tag{C.11b}$$

$$F_2(\Delta_j,a,b) = \text{(ii)}\ [\frac{\Delta_j^2}{2}(b-\alpha_c)+(a-\gamma_2)\Delta_j]\ \ln|\gamma_7| + \frac{(b-\alpha_c)}{2}\cdot$$

$$\Big[(x^2+\frac{\gamma_6^2}{\gamma_7^2})\ln(x^2+\frac{\gamma_6^2}{\gamma_7^2})-x^2\Big]_{x=\gamma_8/\gamma_7}^{\Delta_j+\gamma_8/\gamma_7} + [(a-\gamma_2)$$

$$-\frac{\gamma_8}{\gamma_7}(b-\alpha_c)]\Big[x\ln(x^2+\frac{\gamma_6^2}{\gamma_7^2})-2x+2\frac{\gamma_6}{\gamma_7}\tan^{-1}\frac{x\gamma_7}{\gamma_6}\Big]_{x=\gamma_8/\gamma_7}^{\Delta_j+\gamma_8/\gamma_7}$$

$$\text{if } \gamma_7 \neq 0$$

where $\gamma_6 = (b-\alpha_c)\gamma_1 - (a-\gamma_2)\alpha_s$

$$\gamma_7 = \alpha_s^2 + (b-\alpha_c)^2$$

$$\gamma_8 = (b-\alpha_c)(a-\gamma_2) + \alpha_s\gamma_1$$

$$\gamma_9 = \gamma_1^2 + (a-\gamma_2)^2 \qquad \text{(C.11c)}$$

Finally, the integrals described at the beginning of this section, I_1, I_2, I_3, can be expressed in terms of the more general integral I which in turn is an algebraic combination of closed forms F_1 and F_2. Assume that I is a tunction of its variables and can be written as

$$I = I(\Delta_j,\alpha,a,b,x_j,y_i,\eta_i,x_j,y_j,\eta_j,k_0)$$

Then I_1, I_2 can be written in terms of I as follows.

(1) $I_1 = I(\Delta_j, 0, \Delta_i, 0, x_i, y_i, \eta_i, x_j, y_j, \eta_j, k_0)$

(2) $I_2 = I(\Delta_j, 0, \Delta_i, 0, x_i, -y_i, -\eta_i, x_j, y_j, \eta_j, k_0)$

Assuming $\sin\eta_j \geq 0$ and $\sin\eta_i \geq 0$, I_3 is expressed as

(i) $y_j \geq y_i + \Delta_i \sin\eta_i$

$$I_3 = I(\Delta_j, 0, \Delta_i, 0, x_i, y_i, \eta_i, x_j, y_j, \eta_j, k_0)$$

(ii) $y_i \geq y_j + \Delta_j \sin\eta_j$

$$I_3 = I(\Delta_j, 0, \Delta_i, 0, x_i, y_i, \eta_i, x_j, y_j, \eta_j, -k_0)$$

(iii) $y_i + \Delta_i \sin\eta_i \geq y_j + \Delta_j \sin\eta_j$ and $y_j \geq y_i$ and $\sin\eta_i \neq 0$

$$I_3 = I(\Delta_j, 0, \frac{y_j - y_i}{\sin\eta_i}, \frac{\sin\eta_j}{\sin\eta_i}, x_i, y_i, \eta_i, x_j, y_j, \eta_j, k_0)$$
$$- I(\Delta_j, \Delta_i, \frac{y_j - y_i}{\sin\eta_i}, \frac{\sin\eta_j}{\sin\eta_i}, x_i, y_i, \eta_i, x_j, y_j, \eta_j, -k_0)$$

(iv) $y_i + \Delta_i \sin\eta_i \geq y_j + \Delta_j \sin\eta_j$ and $y_j \geq y_i$ and $\sin\eta_i = 0$

$$I_3 = I(\Delta_j, 0, \Delta_i, 0, x_i, y_i, \eta_i, x_j, y_j, \eta_j, k_0)$$

(v) $y_j + \Delta_j \sin\eta_j \geq y_i + \Delta_i \sin\eta_i$and $y_i \geq y_j$ and $\sin\eta_j \neq 0$

$$I_3 = I(\Delta_i, 0, \frac{y_i - y_j}{\sin\eta_j}, \frac{\sin\eta_i}{\sin\eta_j}, x_i, y_i, \eta_i, x_j, y_j, \eta_j, -k_0)$$
$$- I(\Delta_i, \Delta_j, \frac{y_i - y_j}{\sin\eta_j}, \frac{\sin\eta_i}{\sin\eta_j}, x_i, y_i, \eta_i, x_j, y_j, \eta_j, k_0)$$

(vi) $y_j+\Delta_j\sin\eta_j \geq y_i+\Delta_i\sin\eta_i$ and $y_i \geq y_j$ and $\sin\eta_j = 0$

$$I_3 = I(\Delta_i,0,\Delta_j,0,x_i,y_i,\eta_i,x_j,y_j,\eta_j,k_0)$$

(vii) $y_i+\Delta_i\sin\eta_i > y_j+\Delta_j\sin\eta_j > y_i > y_j$ and $\sin\eta_i \neq 0$ and $\sin\eta_j \neq 0$

$$\begin{aligned} I_3 = {} & I(\frac{y_j-y_i+\Delta_j\sin\eta_j}{\sin\eta_i},0,\frac{y_i-y_j}{\sin\eta_j},\frac{\sin\eta_i}{\sin\eta_j},x_i,y_i,\eta_i,x_j,y_j,\eta_j,-k_0) \\ & - I(\frac{y_j-y_i}{\sin\eta_i}+\Delta_j\frac{\sin\eta_j}{\sin\eta_i},\Delta_j,\frac{y_i-y_j}{\sin\eta_j},\frac{\sin\eta_i}{\sin\eta_j},x_i,y_i,\eta_i,x_j,y_j,\eta_j,k_0) \\ & + I(\Delta_j,0,\Delta_i-\Delta_j\frac{\sin\eta_j}{\sin\eta_i} - \frac{y_j-y_i}{\sin\eta_i},0,x_i,y_i,\eta_i,x_j,y_j,\eta_j,-k_0) \end{aligned}$$

(viii) $y_j+\Delta_j\sin\eta_j > y_i+\Delta_i\sin\eta_i > y_j > y_i$ and $\sin\eta_i \neq 0$ and $\sin\eta_j \neq 0$

$$\begin{aligned} I_3 = {} & I(\frac{y_i-y_j}{\sin\eta_j}+\Delta_i\frac{\sin\eta_i}{\sin\eta_j},0,\frac{y_j-y_i}{\sin\eta_i},\frac{\sin\eta_j}{\sin\eta_i},x_i,y_i,\eta_i,x_j,y_j,\eta_j,k_0) \\ & - I(\frac{y_i-y_j}{\sin\eta_j}+\Delta_i\frac{\sin\eta_i}{\sin\eta_j},\Delta_i,\frac{y_j-y_i}{\sin\eta_i},\frac{\sin\eta_j}{\sin\eta_i},x_i,y_i,\eta_i,x_j,y_j,\eta_j,-k_0) \\ & + I(\Delta_j-\Delta_i\frac{\sin\eta_i}{\sin\eta_j} - \frac{y_i-y_j}{\sin\eta_j},0,\Delta_i,0,x_i,y_i,\eta_i,x_j,y_j,\eta_j,k_0) \end{aligned}$$

With the help of the above closed-form expressions, the evaluation of the kernel matrix elements easily can be obtained.

APPENDIX D

EVALUATION OF THREE-DIMENSIONAL KERNEL MATRIX ELEMENTS

D.1. Introduction

Following the same strategy outlined in Appendix C for the two-dimensional case and employing the same convention in Eq. (C.1,2), we can evaluate the three-dimensional kernel matrix elements in closed form as before. However, only the vertical side wall case is considered here.

D.2. Formulation

As before, in reference to Eq. (2.14), the pulse type basis functions on the surface of the interconnects are chosen as

$$P_j^z(x,y,z) = \left[\begin{array}{l} 1/S_j, \quad \text{for } x_j^- \leq x \leq x_j^+,\ y_j^- \leq y \leq y_j^+,\ z = z_j^o \\ 0, \quad \text{otherwise} \end{array} \right. \tag{D.1a}$$

$$P_j^x(x,y,z) = \left[\begin{array}{l} 1/S_j, \quad \text{for } y_j^- \leq y \leq y_j^+,\ z_j^- \leq z \leq z_j^+,\ x = x_j^o \\ 0, \quad \text{otherwise} \end{array} \right. \tag{D.1b}$$

$$P_j^y(x,y,z) = \left[\begin{array}{l} 1/S_j, \quad \text{for } z_j^- \leq z \leq z_j^+,\ x_j^- \leq x \leq x_j^+,\ y = y_j^o \\ 0, \quad \text{otherwise} \end{array} \right. \tag{D.1c}$$

where S_j is the area of the subdivision, e.g., $S_j = (x_j^+ - x_j^-)(y_j^+ - y_j^-)$ in Eq. (D.1a). Similarly S_j can be obtained for Eq. (D.1b) and Eq. (D.1c). The reason for using S_j as the normalization factor of the pulse amplitude is to have a more stable kernel matrix as discussed in Appendix C. Note that in Eq. (D.1a) x_j^-, x_j^+, y_j^-, y_j^+, are the bounds of the jth rectangular subdivision, and z_j^0 is the ordinate of this "square" on the z-axis. The pulses described in Eq. (D.1) are on the planes parallel to the xy, yz, zx planes, respectively, as the superscripts indicate.

Since the "cores" of the three-dimensional Green's functions in Eqs. (2.23,2.24,2.29) are of the form

$$f(\ [(x-x')^2+(y-y')^2+(z\pm z'+k_0)^2]^{-1/2}\ )$$

some of the typical integrals that result from incorporating them with the pulses in Eq. (D.1) are listed in the following :

$$(1)\quad I_1 = \int_{x_j^-}^{x_j^+} dx \int_{x_i^-}^{x_i^+} dx' \int_{y_j^-}^{y_j^+} dy \int_{y_i^-}^{y_i^+} dy'\ [(x-x')^2+(y-y')^2+(z_j^0-z_i^0+k_0)^2]^{-1/2}$$

$$(2)\quad I_2 = \int_{x_j^-}^{x_j^+} dx \int_{x_i^-}^{x_i^+} dx' \int_{z_j^-}^{z_j^+} dz \int_{z_i^-}^{z_i^+} dz'\ [(x-x')^2+(y_j^0-y_i^0)^2+(z+z'+k_0)^2]^{-1/2}$$

$$(3)\quad I_3 = \int_{x_j^-}^{x_j^+} dx \int_{x_i^-}^{x_i^+} dx' \int_{y_j^-}^{y_j^+} dy \int_{z_i^-}^{z_i^+} dz'\ [(x-x')^2+(y-y_i^0)^2+(z_j^0-z'+k_0)^2]^{-1/2}$$

(4) $$I_4 = \int_{z_j^-}^{z_j^+} dz \int_{z_i^-}^{z_i^+} dz' \int_{x_j^-}^{x_j^+} dx \int_{z_i^-}^{z_i^+} dz' \; [(x-x_i^0)^2+(y_j^0-y)^2+(z-z'+k_0)^2]^{-1/2}$$

where k_0 is a real constant which can also be zero.

Next two closed-form functions are defined such that those integrals presented above can be expressed in terms of them. They are :

$$\begin{aligned} P(\alpha,\beta,\xi) = & -\frac{1}{6}(\alpha^2+\beta^2+\xi^2)^{3/2} + \frac{\xi^2}{2}(\alpha^2+\beta^2+\xi^2)^{1/2} \\ & + \frac{\alpha\beta}{2}(\alpha\sinh^{-1}\frac{\beta}{(\alpha^2+\xi^2)^{1/2}} + \beta\sinh^{-1}\frac{\alpha}{(\beta^2+\xi^2)^{1/2}}) \\ & - \frac{\xi^2}{2}(\beta\sinh^{-1}\frac{\beta}{(\alpha^2+\xi^2)^{1/2}} + \alpha\sinh^{-1}\frac{\alpha}{(\beta^2+\xi^2)^{1/2}}) \\ & - \xi\alpha\beta\tan^{-1}\frac{\alpha\beta}{\xi(\alpha^2+\beta^2+\xi^2)^{1/2}} \end{aligned} \qquad (D.2)$$

$$\begin{aligned} Q(\alpha,\beta,\xi) = & \frac{\alpha\beta}{3}(\alpha^2+\beta^2+\xi^2)^{1/2} - \xi\alpha\beta\sinh^{-1}\frac{\xi}{(\alpha^2+\beta^2)^{1/2}} \\ & - 0.5[\beta(\xi^2-\frac{\beta^2}{3})\sinh^{-1}\frac{\alpha}{(\beta^2+\xi^2)^{1/2}} \\ & +\alpha(\xi^2-\frac{\alpha^2}{3})\sinh^{-1}\frac{\beta}{(\alpha^2+\xi^2)^{1/2}}] \end{aligned}$$

$$+\ 0.5\xi[\ \beta^2\tan^{-1}\frac{\xi\alpha}{\beta(\alpha^2+\beta^2+\xi^2)^{1/2}} + \alpha^2\tan^{-1}\frac{\xi\beta}{\alpha(\alpha^2+\beta^2+\xi^2)^{1/2}}]$$

$$+\ \frac{1}{6}\xi^3\tan^{-1}\frac{\alpha\beta}{\xi(\alpha^2+\beta^2+\xi^2)^{1/2}} \qquad (D.3)$$

It should be pointed out that $P(\alpha,\beta,\xi)$ is an even function on all three variables, while $Q(\alpha,\beta,\xi)$ is an even fuction on the last variable ξ and an odd function on the first two variables α and β.

Finally, adopting the convention in Eqs. (C.1,C.2), we can express the aforementioned integrals as follows :

$$(1)\quad I_1 = \Big[P(\alpha,\beta,z_j^o-z_i^o+k_0)\Big]_{\alpha=x_j^+-x_i^-,\ x_j^--x_i^+,\ \beta=y_j^+-y_i^-,\ y_j^--y_i^+}^{x_j^+-x_i^+\quad x_j^--x_i^-\quad y_j^+-y_i^+\quad y_j^--y_i^-}$$

$$(2)\quad I_2 = \Big[-P(\alpha,\beta,y_j^o-y_i^o)\Big]_{\alpha=x_j^+-x_i^-,\ x_j^--x_i^+,\ \beta=z_j^++z_i^-+k_0,\ z_j^-+z_i^++k_0}^{x_j^+-x_i^+\quad x_j^--x_i^-\quad z_j^++z_i^++k_0\quad z_j^-+z_i^-+k_0}$$

$$(3)\quad I_3 = \Big[-Q(\alpha,\beta,\xi)\Big]_{\alpha=y_j^--y_i^o,\ \beta=z_j^o-z_i^-+k_0,\ \xi=x_j^+-x_i^-,\ x_j^--x_i^+}^{y_j^+-y_i^o\quad z_j^o-z_j^++k_0\quad x_j^+-x_i^+\quad x_j^--x_i^-}$$

$$(4)\quad I_4 = \Big[Q(\alpha,\beta,\xi)\Big]_{\alpha=x_j^--x_i^o,\ \beta=y_j^o-y_i^-,\ \xi=z_j^++z_i^-+k_0,\ z_j^-+z_i^++k_0}^{x_j^+-x_i^o\quad y_j^o-y_i^+\quad z_j^++z_i^++k_0\quad z_j^-+z_i^-+k_0}$$

There are some other possible integrals when incorporating the three-dimensional Green's functions with the pulses in Eq. (D.1), but they are all similar to those just presented. Furthermore, they are all expressible in terms of the two closed-form functions P and Q. Therefore, the kernel matrix elements can all be evaluated by the closed-form expressions described above.

APPENDIX E

SPICE INPUT DECK FOR FULL ADDER I

Shown in the following is the SPICE input deck for "Full Adder Circuit I" with all the interconnect paramters.

```
  full adder circuit I (with cap)
r000     15  16   3.6e+01
r001     16  15   2.0e+01
r002     15  16   5.0e+01
r003     17  18   5.0e+01
r004     17  19   8.6e+01
r005     18  17   7.0e+01
r006     17  18   1.0e+01
c000    500   0   0.0158pf
r007     21 500   1.5e+01
rr007    500  22   1.5e+01
c001    501   0   0.0158pf
r008     21 501   1.5e+01
rr008    501  23   1.5e+01
c002    502   0   0.0158pf
r009     21 502   1.5e+01
rr009    502  24   1.5e+01
c003    503   0   0.0158pf
r010     21 503   1.5e+01
rr010    503  25   1.5e+01
c004    504   0   0.0126pf
r011     26 504   1.5e+01
rr011    504  27   1.5e+01
c005    505   0   0.0126pf
r012     26 505   1.5e+01
rr012    505  28   1.5e+01
c006    506   0   0.0112pf
r013     26 506   2.1e-01
rr013    506  29   2.1e-01
c007    507   0   0.0014pf
r014     29 507   1.5e+01
rr014    507  30   1.5e+01
c008    508   0   0.0126pf
r015     31 508   1.5e+01
rr015    508  32   1.5e+01
c009    509   0   0.0144pf
r016     31 509   5.4e+01
```

```
rr016    509  33    5.4e+01
c010    510   0    0.0126pf
r017     31 510    1.5e+01
rr017    510  34    1.5e+01
c011    511   0    0.0116pf
r018     35 511    3.0e+02
rr018    511  18    3.0e+02
c012    512   0    0.0084pf
r019     36 512    2.7e+01
rr019    512  16    2.7e+01
c013    513   0    0.0087pf
r020     36 513    8.2e+01
rr020    513  37    8.2e+01
c014    504    508    0.0177pf
c015    504    509    0.0211pf
c016    504    510    0.0177pf
c017    505    508    0.0178pf
c018    505    509    0.0211pf
c019    505    510    0.0177pf
c020    506    508    0.0177pf
c021    506    509    0.0211pf
c022    506    510    0.0177pf
c023    507    510    0.0001pf
c024    504    500    0.0210pf
c025    504    501    0.0209pf
c026    504    502    0.0209pf
c027    504    503    0.0209pf
c028    505    500    0.0209pf
c029    505    501    0.0209pf
c030    505    502    0.0209pf
c031    505    503    0.0210pf
c032    506    500    0.0209pf
c033    506    501    0.0209pf
c034    506    502    0.0209pf
c035    506    503    0.0209pf
c036    507    502    0.0001pf
c037    508    513    0.0034pf
c038    509    513    0.0036pf
c039    510    513    0.0034pf
mld000       5   16   16   0    load      l= 12.00um      w=  2.17um
mld001       5   10   10   0    load      l=  8.00um      w=  2.25um
mld002       5   18   18   0    load      l= 17.00um      w=  2.24um
mld003       5   11   11   0    load      l= 10.00um      w=  2.20um
mot000       6   25    0   0    drive     l=  2.00um      w=  6.00um
mot001       7   30    0   0    drive     l=  2.00um      w=  8.00um
mot002       8   23    0   0    drive     l=  2.00um      w=  8.00um
mot003       9   24    0   0    drive     l=  2.00um      w=  8.00um
mot004      16   30    9   0    drive     l=  2.00um      w=  8.00um
mot005      16   34    8   0    drive     l=  2.00um      w=  8.00um
mot006      15   34    7   0    drive     l=  2.00um      w=  8.00um
mot007      10   16    0   0    drive     l=  2.00um      w=  8.00um
```

```
mot008    11  35   0  0   drive     l=  2.00um      w=  8.00um
mot009    12  27   0  0   drive     l=  2.00um      w=  8.00um
mot010    18  32  20  0   drive     l=  2.00um      w=  6.00um
mot011    17  37  12  0   drive     l=  2.00um      w=  8.00um
mot012    18  37  14  0   drive     l=  2.00um      w=  8.00um
mot013    20  28   6  0   drive     l=  2.00um      w=  6.00um
mot014    13  33   0  0   drive     l=  2.00um      w=  8.00um
mot015    19  37  13  0   drive     l=  2.00um      w=  8.00um
mot016    14  22   0  0   drive     l=  2.00um      w=  8.00um
*load capacitances
* sum
cl1  11 0  .1pf
* carry
cl2  10 0  .1pf
.model load nmos vto=-2. uo=600 level=1 lambda=0.02 tox=1.e-7
.model pass nmos vto=1.  uo=600 level=1 lambda=0.02 tox=1.e-7
.model drive nmos vto=1. uo=600 level=1 lambda=0.02 tox=1.e-7
*inputs
v3  32  0  pulse(0 5 120n 0.1n 0.1n 120n 240n)
v2  28  0  pulse(0 5 60n 0.1n 0.1n 60n 120n)
v1  25  0  pulse(0 5 30n 0.1n 0.1n 30n 60n)
vdd 5 0 5
.options lvltim=1 itl4=20
.tran 3n 240n
.print tran v(25) v(28) v(32) v(11) v(10)
.end
```

Note that the capacitors numbered from "c014" to "c039" are coupling capacitances. They are "commented out" in the case of only considering the self-capacitances. For the case of no interconnect parameters, another extraction is done with large parameters "RLIMIT" and "CLIMIT" specified in "constant.h".

APPENDIX F

SPICE INPUT DECK FOR FULL ADDER II

The input SPICE deck for "Full Adder Circuit II" is given in the following.

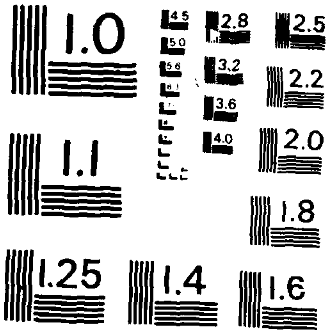
```
 full adder circuit II (with capacitance)
r000     10   7   3.5e+01
r002     12  13   6.9e+01
r003     12  14   1.0e+01
c000    500   0   0.0114pf
r004     14 500   5.1e+00
rr004    500  15   5.1e+00
c001    501   0   0.0109pf
r005     14 501   5.1e+00
rr005    501  16   5.1e+00
c002    502   0   0.0114pf
r006     17 502   4.9e+01
rr006    502  18   4.9e+01
c003    503   0   0.0335pf
r007     17 503   5.4e+01
rr007    503  19   5.4e+01
c004    504   0   0.0114pf
r008     17 504   8.6e-02
rr008    504  20   8.6e-02
r009     20  21   1.0e+01
c005    505   0   0.0316pf
r010     22 505   5.2e+00
rr010    505   8   5.2e+00
r011     23  24   5.0e+01
r012     23  25   1.1e+02
r013     23  26   1.0e+02
r014     27  28   3.0e+01
r015     29  28   5.5e+01
r016     30  31   2.5e+01
c006    506   0   0.0427pf
r017     32 506   2.0e+02
rr017    506  33   2.0e+02
c007    507   0   0.0339pf
r018     34 507   3.5e+02
rr018    507   6   3.5e+02
c008    508   0   0.0043pf
r019     35 508   4.5e+01
```

```
rr019    508  31    4.5e+01
c009    509   0    0.0144pf
r020     36 509    1.5e+01
rr020    509  37    1.5e+01
c010    510   0    0.0193pf
r021     36 510    1.5e+01
rr021    510  38    1.5e+01
c011    511   0    0.0370pf
r022     36 511    3.2e+02
rr022    511  39    3.2e+02
c012    512   0    0.0065pf
r023     40 512    6.8e+01
rr023    512  41    6.8e+01
c013    513   0    0.0289pf
r024     40 513    3.0e+02
rr024    513  42    3.0e+02
c014    514   0    0.0101pf
r025     40 514    1.1e+02
rr025    514  25    1.1e+02
c015    515   0    0.0296pf
r026     43 515    3.1e+02
rr026    515  44    3.1e+02
c016    516   0    0.0279pf
r027     45 516    9.0e+01
rr027    516  46    9.0e+01
c017    517   0    0.0179pf
r028     42 517    3.0e+01
rr028    517  47    3.0e+01
c018    518   0    0.0173pf
r029     44 518    1.8e+02
rr029    518  28    1.8e+02
c019    519   0    0.0065pf
r030     37 519    6.8e+01
rr030    519   7    6.8e+01
c020    505    503    0.0011pf
c021    505    506    0.0222pf
c022    505    507    0.0222pf
c023    505    512    0.0229pf
c024    505    513    0.0229pf
c025    505    514    0.0265pf
c026    503    506    0.0285pf
c027    503    507    0.0285pf
c028    502    511    0.0030pf
c029    502    513    0.0059pf
c030    503    513    0.0314pf
c031    504    513    0.0030pf
c032    516    507    0.0120pf
c033    506    507    0.0096pf
c034    510    518    0.0175pf
c035    511    517    0.0052pf
c036    517    515    0.0052pf
```

```
c037   500   515   0.0030pf
c038   501   515   0.0030pf
mld000      5  31  31  0   load       l= 14.00um     w=  5.71um
mld001      5   6   6  0   load       l= 14.00um     w=  5.71um
mld002      5   8   8  0   load       l= 14.00um     w=  5.71um
mld003      5   7   7  0   load       l= 14.00um     w=  5.71um
mld004      5  25  25  0   load       l= 14.00um     w=  5.71um
mld005      5  28  28  0   load       l= 12.00um     w=  6.00um
mot000      6  32   0  0   drive      l=  4.00um     w=  8.00um
mot001     10  34   0  0   drive      l=  4.00um     w=  8.00um
mot002      7  35   0  0   drive      l=  4.00um     w=  8.00um
mot003      8  19   0  0   drive      l=  4.00um     w=  8.00um
mot004     21  39  16  0   pass       l=  4.00um     w=  4.00um
mot005     15  41  21  0   pass       l=  4.00um     w=  4.00um
mot006     13  43  22  0   pass       l=  4.00um     w=  4.00um
mot007     26  45   0  0   drive      l=  4.00um     w=  8.00um
mot008     24  33   0  0   drive      l=  4.00um     w=  8.00um
mot009     27  42   0  0   drive      l=  4.00um     w= 12.00um
mot010      5  38   9  0   pass       l=  4.00um     w=  4.00um
mot011     18  44   9  0   drive      l=  4.00um     w=  4.00um
mot012      9  47   0  0   drive      l=  4.00um     w=  4.00um
mot013     29  37   0  0   drive      l=  4.00um     w=  8.00um
mot014     30  46   0  0   drive      l=  4.00um     w=  8.00um
*load capacitances
* carry
cl1  9 0  .1pf
* sum
cl2  14 0  .1pf
* model cards
.model load nmos vto=-3. uo=600 level=1 lambda=0.02 tox=1.e-7
.model pass nmos vto=1.  uo=600 level=1 lambda=0.02 tox=1.e-7
.model drive nmos vto=1. uo=600 level=1 lambda=0.02 tox=1.e-7
*inputs
v3  19  0  pulse(0 5 120n 0.1n 0.1n 120n 240n)
v2  46  0  pulse(0 5 60n 0.1n 0.1n 60n 120n)
v1  32  0  pulse(0 5 30n 0.1n 0.1n 30n 60n)
vdd 5 0 5
.tran 3n 240n
.print tran v(32) v(46) v(19) v(14) v(9)
.end
```

Note that the capacitances from c020 to c038 are coupling capacitances. Some of the interconnect parameters consist of resistances only, e.g., r011 to r016, because the capacitances of diffusion interconnect lines are neglected in the consideration. Other considerations are the same as in the previous case.

AD-A142 381 MODELING AND EXTRACTION OF INTERCONNECT PARAMETERS IN 3/3
VERY-LARGE-SCALE INTEGRATED CIRCUITS(U) ILLINOIS UNIV
AT URBANA COORDINATED SCIENCE LAB C P YUAN AUG 83
UNCLASSIFIED R-994 N00014-79-C-0424 F/G 9/5 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

REFERENCES

[1] "Special Issue on VLSI Design : Problems and Tools", IEEE Proceedings, Jan. 1983.

[2] W. H. Dierking and J. D. Bastian, "VLSI Parasitic Capacitance Determination by Flux Tubes", 24th Midwest Symposium of Circuits and Systems, Univ. of New Mexico, Albuquerque, June 29, 1981.

[3] R. L. M. Dang and N. Shigyo, "Coupling Capacitances for Two-Dimensional Wires", IEEE Electron Device Letters, vol. EDL-2, no. 8, pp. 196-197, Aug. 1981.

[4] W. T. Weeks, "Calcualtion of Coefficients of Capacitance of Multiconductor Transmission Lines in the Presence of a Dielectric Interface", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-18, no. 1, pp. 35-43, Jan. 1970.

[5] N. G. Alexopoulos, J. A. Maupin and P. T. Greiling, "Determination of the Electrode Capacitance Matrix for GaAs FET's", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-28, no. 5, pp. 459-466, May 1980.

[6] P. D. Patel, "Calculation of Capacitance Coefficients for A System of Irregular Finite Conductors on a Dielectric Sheet", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-19, no. 11, pp. 862-869, Nov. 1971.

[7] P. E. Cottrell, E. M. Buturla and D. R. Thomas, "Multi-Dimensional Simulation of VLSI Wiring Capacitance", IEEE International Electron Devices Meeting Proceedings, San Francisco, CA., Dec. 1982.

[8] A. E. Ruehli and P. A. Brennan, "Capacitance Models for Integrated Circuit Metallization Wires", IEEE J. of Solid-State Circuits, vol. SC-10, no. 6, pp. 530-536, Dec. 1975.

[9] A. E. Ruehli, "Equivalent Circuit Models for Three-Dimensional Multiconductor Systems", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-22, no. 3, pp. 216-220, March 1974.

[10] A. E. Ruehli and P. A. Brennan, "Efficient Capacitance Calculations for Three-Dimensional Multicnductor Systems", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-21, no. 2, pp. 76-82, Feb. 1973.

[11] R. F. Harrington, Field Computation by Moment Methods, R. E. Krieger Publishing, Malabar, Florida, 1982.

[12] G. Bilardi, M. Pracchi and F. P. Preparata, "A critique of Network Speed in VLSI Models of Computation", IEEE J. of Solid-State Circuits, vol. SC-17, no. 4, pp. 696-702, Aug. 1982.

[13] A. E. Ruehli, "Survey of Computer-Aided Electrical Analysis of Integrated Circuit Interconnections", IBM J. of Research and Development, vol. 23, no. 6, pp. 626-639, Nov. 1979.

[14] S. Ramo, J. R. Whinnery and T. V. Duzer, Fields and Waves in Communication Electronics, John Wiley and Sons, New York, 1967.

[15] W. R. Smythe, Static and Dynamic Electricity, McGraw-Hill, New York, 1950.

[16] M. Abramowitz and I. A. Stegun, Handbook of Mathematical Functions, Dover, New York, 1970.

[17] I. S. Gradshteyn and I. M. Ryzhik, Table of Integrals, Series and Products, Academic Press, New York, 1980.

[18] R. W. Keyes, "The Evolution of Digital Electronics Towards VLSI", IEEE Trans. on Electron Devices, vol. ED-26, no. 4, pp. 271-279, Apr. 1979.

[19] A. Farrar and A. T. Adams, "Multilayer Microstrip Transmission Lines", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-22, no. 10, pp. 889-891, Oct. 1974.

[20] H. B. Palmer, "The Computation of a Parallel Plate Capacitor by the Schwartz-Christoffel Transformation", Trans. of AIEE, vol. 56, no. 3, pp. 363-366, March 1937.

[21] W. H. Chang, "Analytic IC Metal-Line Capacitance Formulas", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-24, no. 9, pp. 608-611, Sept. 1976; and correction in vol. MTT-25, no. 8, p.712, Aug. 1977.

[22] M. I. Elmasry, "Capacitance Calculations in MOSFET VLSI", IEEE Electron Device Letters, vol. EDL-3, no. 1, pp. 6-7, Jan. 1982.

[23] C. P. Yuan and T. N. Trick, "A Simple Formula for the Estimation of the Capacitance of Two-Dimensional Interconnects in VLSI Circuits", IEEE Electron Device Letters, vol. EDL-3, no. 12,

pp. 391-393, Dec. 1982.

[24] T. Sakurai and K. Tamaru, "Simple Formulas for Two- and Three-Dimensional Capacitances", IEEE Trans. on Electron Devices, vol. ED-30, no. 2, pp. 183-185, Feb. 1983.

[25] R. Shrivastava and K. Fitzpatrick, "A Simple Model for the Overlap Capacitance of a VLSI MOS Device", IEEE Trans. on Electron Devices, vol. ED-29, no. 12, Dec. 1982.

[26] J. L. Bentley, "Multidimensional Binary Search Trees Used for Associative Searching", Commun. of the Ass. Comput. Mach., vol. 18, no. 9, pp. 509-517, Sept. 1975.

[27] U. Lauther, "4-Dimensional Binary Search Trees as a Means to Speed up Associative Searches in Design Rule Verification of Integrated Circuits", Design Automation and Fault-Tolerant Computing, vol. 2, no. 3, pp. 241-247, July 1978.

[28] P. H. Munch and K. H. Munch, "A General Solution for Design Verification", IEEE International Conference on Circuits and Computers Proceedings, pp. 320-323, New York, 1982.

[29] C. P. Yuan and T. N. Trick, "Extraction of Interconnect Parameters in VLSI Circuits", IEEE International Symp. on Circuits and Systems Proceedings, pp. 1229-1232, May 1983.

[30] R. L. Rivest, "Analysis of Associative Retrieval Algorithms", Technical Report STAN-CS-74-415, Computer Science Department, Stanford University, May 1974.

[31] A. V. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974.

[32] T. Mitsuhashi, T. Chiba and M. Takashima, "An Integrated Mask Artwork Analysis System", 17th Design Automation Conference Proceedings, pp. 277-284, 1980.

[33] P. Losleben and K. Thompson, "Topological Analysis for VLSI Circuits", 16th Design Automation Conference Proceedings, pp. 461-473, 1979.

[34] H. Baird and Y. Cho, "An Artwork Design Verification System", 12th Design Automation Conference Proceedings, pp. 414-420, 1975.

[35] B. Preas, B. Lindsay and C. Gwyn, "Automatic Circuit Analysis Based on Mask Information", 13th Design Automation Conference Proceedings, pp. 309-317, 1976.

[36] K. Yoshida, et al., "A Layout Checking System for Large Scale Integrated Circuits", 14th Design Automation Conference Proceedings, pp. 322-320, 1977.

[37] M. E. Daniel and C. W. Gwyn, "CAD Systems for IC Design", IEEE Trans. on CAD of IC's and Systems, vol. CAD-1, no. 1, pp. 2-12, Jan. 1982.

[38] C. Mead and L. Conway, Introduction to VLSI Systems, Addison-Wesley, Reading, Mass., 1980.

[39] D. E. Knuth, The Art of Computer Programming, vol. 1, Addison-Wesley, Reading, Mass., 1973.

[40] M. Graf, private communication

[41] R. E. Collin, Field Theory of Guided Waves, McGraw-Hill, New York, 1960.

[42] K. G. Black and T. J. Higgins, "Rigorous Determination of the Parameters of Microstrip Transmission Lines", IRE Trans. on Microwave Theory and Techniques, vol. MTT-3, no. 3, pp. 93-113, March 1955.

[43] J. W. Duncan, "Characteristic Impedances of Multiconductor Strip Transmission Lines", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-13, no. 1, pp. 107-118, Jan. 1965.

[44] H. Guckel, "Characteristic Impedance of Generalized Rectangular Transmission Lines", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-13, no. 5, pp. 270-274, May 1965.

[45] E. Yamashita and R. Mittra, "Variational Method for the Analysis of Microtrip Lines", IEEE Trans. on Microwave Theory and Techniques, vol. MTT-16, no. 4, pp. 251-256, Apr. 1968.

[46] E. C. Jordan and K. G. Balmain, Electromagnetic Waves and Radiating Systems, Prentice-Hall, Englewood Cliffs, New Jersey, 1968.

[47] A. K. Sinha, J. A. Cooper, Jr., and J. J. Levinstein, "Speed Limitations Due to Interconnect Time Constants in VLSI Integrated Circuits", IEEE Electron Device Letters, vol. EDL-3, no. 4,

pp. 90-92, Apr. 1982.

[48] P. Yang, "An Investigation of Ordering, Tearing, and Latency Algorithms for the Time-Domain Simulation of Large Circuits", CSL Report R-891, University of Illinois at Urbana-Champaign, August 1980.

[49] S. A. Sohail, "A Simulation Program with Latency Exploitation for the Transient Analysis of Digital Circuits", M. S. Thesis, University of Illiois at Urbana-Champaign, 1983.

VITA

Chen-Ping Yuan was born in Taipei, Taiwan, Republic of China on January 11, 1953. He attended National Taiwan University in Taipei and received his B.S. degree in Electrical Engineering in June 1975. He was a second lieutenant in the Chinese Army from 1975 to 1977. During this period, he taught English and basic electronics to NCO's. In August 1977 he entered the University of Illinois at Urbana-Champaign on a research assistantship under the direction of Professors C. H. Liu and K. C. Yeh in the Ionospheric Radio Lab. From 1978 to 1980 he was a research assistant for Professor Y. T. Lo in the Electromagnetics Lab. He has held a teaching assistantship with the Electrical Engineering Department or a research assistantship with the Coordinated Science Lab for Professor T. N. Trick since then. His research interests are in the areas of computer-aided design, VLSI circuits, design verification, numerical analysis, and semiconductor device modeling.

END

FILMED

DTIC