able to map each input noncode word to an output noncode word, and thus, the SCD property of the whole checker is not lost.

Examples illustrating the points a, b, and c have been given in another report [17].

V. Conclusions

The paper discusses some important aspects of self-checking checkers and highlights some problems concerning the necessity of the fault secure property for SCD checkers. Some authors have claimed that in order to ensure the exercising of the final checker in a complex self-checking system, the partial self-testing checkers need the fault secure property and some other authors claimed that if the partial checkers are SCD, the SFS property is needed as well. Clearly the fault secure property is not useful for self-testing checkers. For SCD checkers this property is not useful for the exercising of the partial checkers. For the exercising of the final checker this property is neither necessary nor sufficient. It will be useless in the following cases:

- the exercising of the final checker is based on the nondependency of the data,
- there is a complete or a partial data dependency but the exercising is ensured by the technique given in the Section 4.1,
- a self-exercising checker [14], [15] is used as final double-rail checker.

The SFS property may be useful in case the exercising of the final checker is established by simulation of the whole self-checking system under the fault free condition. In some cases this technique can be used successfully. But in several other cases this simulation can be too complex indeed unpractical even for the SCD/SFS checkers. It is also possible that the system does never generate a vector set which exercises the final checker and of course the simulation cannot ensure the final checker exercising (it just can be used to check if this exercising is ensured). Then, the exercising of the final checker must be ensured by one of the previously proposed techniques and the SFS property of the partial checkers is not useful. The conclusion is that the basic property for checkers is the SCD property (which includes the self-testing property as a particular case), any checker design must verify it.

References

[1] D. A. Anderson, "Design of self-checking digital networks using coding techniques," Rep. 527, CSL/University of Illinois, Urbana, Sept. 1971.
[2] W. C. Carter and P. R. Schneider, "Design of dynamically checked computers," in *Proc. IFIP Congress*, Edinburgh, U.K., 1968.
[3] M. Nicolaidis, I. Jansch, and B. Courtois, "Strongly code disjoint checkers," in *Proc. 14th Int. Symp. on Fault Tolerant Computing*, Kissimmee, FL, June 20–22, 1984, pp. 16–21.
[4] Y. Tamir and C. H. Sequin, "Design and application of self-testing comparators implemented with MOS PLA's," *IEEE Trans. Comput.*, vol. C-33, pp. 493–506, June 1984.
[5] A. M. Paschalis, D. Nikolos, and C. Halatsis, "Efficient modular design of TSC checkers for m-out-of-2m codes," *IEEE Trans. Comput.*, vol. C-37, pp. 301–309, Mar. 1988.
[6] N. K. Jha, "Strongly fault-secure and strongly self-checking domino-CMOS implementations of totally self-checking circuits," *IEEE Trans. Computer-Aided Design*, vol. 9, Mar. 1990.
[7] D. A. Anderson and G. Metze, "Design of totally self-checking check circuits for m-out-of-n codes," *IEEE Trans. Comput.*, vol. C-22, pp. 263–269, Mar. 1973.
[8] M. A. Marouf and A. D. Friedman, "Design of self-checking checkers for Berger codes," in *Proc. FTCS-8*, 1978, pp. 179–184.
[9] J. E. Smith and G. Metze, "Strongly fault secure logic networks," *IEEE Trans. Comput.*, June 1978.
[10] N. K. Jha, "Fault detection in CVS parity trees: Application to SSC CVS parity and two-rail checkers," in *Proc. 19th Int. Symp. Fault Tolerant Computing*, Chicago, IL, June 21–23, 1989.
[11] T. Nanya and T. Kawamura, "Error secure/propagating concept and its application to the design of strongly fault secure processor," *IEEE Trans. Comput.*, vol. C-37, pp. 14–24, Jan. 1988.
[12] T. Nanya and M. Uchida, "A strongly fault-secure and strongly code-disjoint realization of combinational circuits," in *Proc. 19th Int. Symp. Fault Tolerant Computing*, Chicago, IL, June 21–23, 1989.
[13] M. Nicolaidis and B. Courtois, "Strongly code-disjoint checkers," *IEEE Trans. on Comput.*, vol. C-37, pp. 751–756, June 1988.
[14] M. Nicolaidis, "A unified built-in-self-test scheme: UBIST," in *Proc. 18th Int. Symp. Fault Tolerant Computing*, Tokyo, Japan, June 27–30, 1988.
[15] M. Nicolaidis, "Self-exercising checkers for unified built-in self-test (UBIST)," *IEEE Trans. Computer-Aided Design*, vol. 8, pp. 1203–218, Mar. 1989.
[16] E. Fujiwara and K. Matsuoka, "A totally self-checking generalized prediction checker and its application for built-in testing," in *Proc. 15th Int. Symp. Fault Tolerant Computing*, Ann Arbor, MI, June 19–21, 1985.
[17] M. Nicolaidis, "Basic properties of strongly code-disjoint checkers," IMAG Rep. RR 824-I-, Sept. 1990.
[18] M. Boudjit and M. Nicolaidis, "A tool for computing the output code spaces and verifying the self-checking properties in complex self-checking systems," *IEICE Trans. Inf. & Syst.*, vol. E75-D, no. 6, pp. 824–834, Nov. 1992.
[19] S. Kundu and S. M. Reddy, "Embedded self-checking checkers: A practical design," *IEEE Design & Test*, Aug. 1990.

Hierarchical Analysis of High Frequency Interconnect Networks

Janusz A. Starzyk

Abstract— **Interconnect networks are analyzed using symbolic frequency domain analysis and an exact model of a distributed line. The analyzed network must have a tree structure and may contain transmission lines as well as other linear two-ports with specified transmission matrices. Any regular portion of the interconnect network is analyzed hierarchically with significant savings in analysis time. Numerical Laplace transform inversion is used to obtain time domain solutions. The event driven approach is adopted to reduce analysis time. Time complexity of the method depends on the regularity of the analyzed network, and can be as low as a logarithmic function of the number of elements for a hierarchically organized tree structure.**

I. Introduction

Transmission line effects were not a serious concern when signal wavelength was much larger than the dimensions of the designed digital circuits. But in recent years, in circuits with finer features and higher signal frequencies, transmission line effects are of concern at all interconnections [1]. In the advanced BiMOS/CMOS circuits, the signal rise time is comparable to propagation delays, and transmission line behavior has to be considered.

Manuscript received October 21, 1991; revised June 18, 1993. This paper was recommended by Associate Editor J. White.

The author was with AT&T Bell Laboratories, VLSI Systems Research Lab., Holmdel, NJ. He is now with the Department of Electrical and Computer Engineering, Ohio University, Athens, OH 45701.

IEEE Log Number 9214739.

Timing analyzers tried to deal with the analysis of interconnections by using simplified models, usually RC trees, and either estimating the interconnect delay [2]–[5] or approximating its response [6]–[11]. Estimation of the delay is simpler, and can be done with a reasonable accuracy and significant reduction in the analysis time. However, as signal frequencies are being increased, there is a growing need to consider the analog behavior, due to the increasing effect of line inductances.

In [5] exact r-c-g models of distributed lines were used, in order to improve the accuracy of the predicted time delays. Accuracy of the RC ladder approximation of a distributed line was studied in [12], [13] and, in general, depends on the type of material used to build a distributed line, as well as the number and type of sections in the ladder approximation. Another tendency in the interconnect analysis, is a consideration of the line inductance and the coupling capacitance, in order to evaluate waveforms of the response which, if applied to a digital circuit, may cause unintended switching. This behavior may go undetected by methods based on the RC trees, even when the double time-constant models are used [14].

The asymptotic waveform evaluation method presented in [6] satisfies most of the requirements of high frequency interconnect network analysis. It approximates the response of a circuit with floating capacitors, grounded resistors, inductors, and controlled sources. Its accuracy depends on the number of moments used, and the method is equivalent to RC tree methods if only the first moments are used. The method is from one to two orders of magnitude faster than HSPICE. A generalization of the asymptotic waveform evaluation method was presented in [10], where authors introduced the moment polynomial nodal analysis combined with the nodal analysis to solve the interconnect networks. This generalization permits the analysis of networks with distributed elements. Different generalization of the asymptotic waveform evaluation, presented in [11], included lossy coupled transmission lines and nonlinear loads.

Another type of approximation in the interconnect analysis was used in [9], where driving point admittance of an RC tree was approximated in order to improve the simulation accuracy. A noteworthy approach was developed in [15], where state equations in the complex frequency domain and the inverse Laplace transform were used to obtain time domain solutions. The inverse Fourier transformation of the frequency domain scattering parameters was used in [16] to determine the impulse response of transmission lines. Recently, Pade approximations of the transmission line's characteristic admittance and the exponential propagation function were used to derive a recursive convolution formulation [17]. This method can be applied to lossy transmission lines terminated with nonlinear elements and is one to two orders of magnitude faster than Spice3.e.

In this paper, a method for the analysis of a tree of transmission lines and other two-ports is presented. Exact models are used for transmission lines with all r-l-c-g parameters specified. This way any approximation introduced by discrete models of transmission lines is avoided. Discrete elements can be inserted between transmission lines and may contain floating capacitors, grounded resistors, inductors, and controlled sources. If the interconnect network has a regular structure, which is often the case in VLSI design, it may be represented in a hierarchical form. The method uses network equations in the complex frequency domain and an event-driven approach to compute the inverse Laplace transform only when the response changes more than a prespecified limit. In addition, a hierarchical analysis is used in order to reduce simulation time in regular structures. As a result, a hierarchically organized interconnect network can be analyzed in logarithmic time. This method is very efficient, particularly when hierarchical structures are present, and produces accurate results.

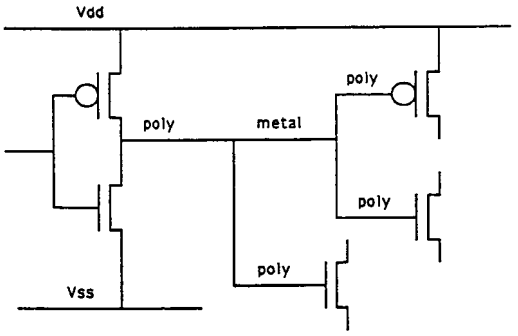

Fig. 1. A simple interconnect network.

The paper is organized as follows. First, the problem and the main objective of this work are stated. Errors related to the discrete approximation of distributed lines are briefly discussed. The main steps of the interconnect network analysis method are presented in Section III. In the next section the concept of hierarchical analysis is introduced and the basic steps of the related algorithms are explained. Then, in Section V, time domain solution based on the inverse Laplace transform is discussed. Finally, in Section VI, computational complexity of the method is discussed, and its time requirements are compared with the requirements of the Advice program.

II. Statement of the Problem

The interconnect networks considered in this paper are tree structures, which contain linear two-ports described by their transmission matrices. In particular, transmission lines can be represented exactly by the complex frequency domain solutions of differential equations which relate line voltages and currents. Using these solutions, the lines' transmission matrices can be formulated. Lumped elements, like floating capacitors and inductors, can be inserted at any place of the tree structure. One of the nonground nodes is assumed to be the input, and one or more are the outputs. A simple interconnect network is shown in Fig. 1.

Much more complex structures of interconnect networks are used in VLSI designs. These structures—such as clock and control lines—show a large degree of regularity in hierarchically organized designs. Identification of such hierarchically organized structures is facilitated in VLSI design, where hierarchical design tools are used, and where many subsystems have regular, cellular organization.

Binary trees are used as the basic structures. Other networks are converted to this format before analysis. Such conversion is simply realized by the insertion of two-ports having the unit transmission matrices. Since binary trees have simple, regular structures, the algorithms which use these trees are fast, and have a simple source code.

The problem is to find network responses on the specified outputs to any signal for which the Laplace transform can be determined. While many techniques exist to solve this problem, the aim is to do it in a minimum time. The reason for focusing the effort on the analysis time is the ever growing size and complexity of VLSI circuits and a need to perform interconnect network analysis more efficiently.

III. Analysis of the Tree Networks

The basic structure of the interconnect networks considered in this paper is a binary tree with branches being either transmission lines, discrete two-ports, or hierarchically organized structures of

transmission lines. While a discrete approximation of distributed elements is widely used and can give a good approximation of the resulting signals [2]–[9], [12], [13], a direct representation of transmission lines is used here in order to reduce computations and improve waveform accuracy.

Sakurai [12] studied the accuracy of approximation of the transmission line delay by the RC ladder using different numbers of RC sections and different topologies. He showed that a relatively large number of Γ sections is required to obtain a small time delay error. For instance, eight Γ sections are needed to reduce the error below 10%, and more than 15 such sections are needed to reduce the error below 5%. Much better results can be obtained if T or Π sections are used where the required number of sections is smaller. Selection of the optimum number of RC sections needed to approximate a distributed RC element for a given signal bandwidth was given in [13]. The accuracy, in general, depends on the load, network topology, and the number of elements used.

In addition to the delay errors, significant waveform deviations may occur. These errors may be more important than timing errors, if additional switching is overlooked as a result of using discrete models. Since the method discussed in this paper uses exact models of transmission lines, the only errors in the interconnect network analysis are the numerical errors related to the inverse Laplace transform, and roundoff errors, which depend on the machine's accuracy. Errors of the numerical inversion can be minimized by increasing the accuracy of the procedure used for the inverse Laplace transform.

Analysis of a tree network is performed first in the complex frequency domain. Next, time domain solutions are obtained by using the event driven inverse Laplace transform. The remaining part of this section and the following section are devoted to the s-domain analysis. Time domain analysis is discussed in Section V.

3.1. Frequency Domain Analysis

All the two ports in the network are presented by their transmission matrices which relate input and output variables on two-port terminals. In particular, a transmission line is represented by a matrix

$$T = \begin{bmatrix} \cosh(\gamma l) & Z_f \sinh(\gamma l) \\ \dfrac{\sinh(\gamma l)}{Z_f} & \cosh(\gamma l) \end{bmatrix} \tag{1}$$

where $\gamma = \sqrt{(R+sL)(G+sC)}$, and $Z_f = \sqrt{(R+sL)/(G+sC)}$, l is the length of the line, s if the complex frequency (variable in the Laplace transform), and R, L, C, and G are per unit length line parameters. A single horizontal element (e.g., floating capacitor) is represented by a two-port described by

$$T = \begin{bmatrix} 1 & Z(s) \\ 0 & 1 \end{bmatrix} \tag{2}$$

where $Z(s)$ is the impedance of this discrete element at a given complex frequency, and a single vertical element is represented by

$$T = \begin{bmatrix} 1 & 0 \\ Y(s) & 1 \end{bmatrix} \tag{3}$$

Other two-ports may be used as long as their transmission matrices are defined as functions of the complex frequency. Transmission matrices of all branches are evaluated during the first step of interconnect network analysis.

The hierarchical structures in this paper are defined in terms of the composite transmission lines, and are analyzed in Section IV. A result of this analysis is a transmission matrix, which describes the hierarchical structure, and is evaluated at a single complex frequency s.

Next, the load and input admittances are computed at each tree node. The load admittance of a tree node is obtained by adding the input admittances of the branches connected to this node and the admittance of a discrete load at this node. The input admittance of a branch is a function of loads and two-port parameters of the entire subtree structure connected to its output. All these admittances can be evaluated very efficiently using a recursive tree folding procedure. The procedure is fairly straightforward and is omitted for simplicity of the presentation.

As the last stage of the complex frequency domain analysis, the required transfer functions from the root to the specific outputs are computed. This analysis is also straightforward and uses a property of the cascade connection of two-ports. The transmission matrix of such a connection equals to the product of transmission matrices of the constitute two-ports. Such a product can be obtained very efficiently for identical two-ports.

3.2. Algorithm

Let T_i denote the transmission matrix of the branch B_i. Let Y_i be a load admittance of the ith node of the tree, and Y_{in_i} be the input admittance of the branch B_i. The basic steps of the algorithm for analysis of the tree networks can be summarized as follows:

1) Compute the transmission matrix of each branch.
2) Compute the discrete load admittance at each node.
3) Compute the input admittance of each branch.
4) Compute the load admittance of each node.
5) For each output find a transmission matrix

$$T = T_{i_1} T_{ld_{i_1}} T_{i_2} T_{ld_{i2}} \cdots T_{i_{\text{out}}}$$

where $i_1, i_2, \cdots, i_{\text{out}}$ are the indices of branches on the unique path from the root to the output node, and T_{ld_i} is the transmission matrix, which represents the load effect of the stray branches at the ith node of the tree (a stray branch is a branch not included in the unique path from a given output node to the root). Matrix T_{ld_i} is computed as

$$T_{ld_i} = \begin{bmatrix} 1 & 0 \\ Y_{ld_i} & 1 \end{bmatrix} \tag{4}$$

and $Y_{ld_i} = Y_i - Y_{\text{in}_{i\_\text{next}}}$, where i_next stands for the index of the next branch in the path from the input to the output.

6) For each output find the transfer function

$$\frac{V_{\text{out}}}{V_{\text{in}}} = \frac{1}{A + BY_{\text{out}}}$$

where A and B are the elements of T.

Each of these steps takes time proportional to either the number of nodes or the number of branches in the tree. Since the number of branches in a tree is equal to the number of nodes minus one, we see that this algorithm can be completed in time which is a linear function of the number of branches.

IV. Hierarchical Analysis

This section presents hierarchical analysis of the interconnect network, in which at least one branch is described by a *hierarchically organized structure* (HOS). A HOS is an interconnect structure of identical cells. Each cell may contain transmission lines, discrete two-ports and HOS's. If a cell contains no HOS then it is considered a simple branch, otherwise the hierarchy level of the HOS is increased by one. This kind of structure corresponds to a hierarchically designed VLSI subcircuit which contains an array of identical cells, each composed of another array of lower level cells. Each HOS is described in the analysis program in its unexpanded form for easier identification and for hierarchical analysis.

The concept of hierarchical analysis is based on two observations. The first observation relates to the identical branches. If the branches are identical, then only one of them has to be analyzed. All such branches have identical transmission matrices. The second observation relates to the identical subtrees. The input admittances of such subtrees are equal. Using these two observations, considerable processing time is saved, which otherwise would be spent on the simulation of identical structures.

Let us compare the computing effort needed to analyze a single HOS with a similar effort for the flat network analysis. Assume first, that a single level of a hierarchical structure is analyzed, and that the two-ports, which represent the subcircuits at this level, are identical and connected in a specific, regular way. A typical, regular connection of two-ports on a single hierarchy level is the cascade connection. Other connections, for instance serial or parallel connection of two-ports, are possible. The only requirement is that they are repeated in every substructure on the same level of hierarchy.

These regular connections of two-ports are analyzed very efficiently. For instance, the cascade connection of identical two ports can be analyzed in constant time, independent of the number of cascaded two-ports. This analysis is performed using the eigenvalues and the characteristic equation of the transmission matrices describing the cascaded two-ports. A detailed discussion of this type of analysis is presented in the Appendix.

The analysis time for the same structure in the flat network is, in the best case, proportional to the number of sections. Therefore, the analysis which recognizes the regularity of this structure, is up to n times faster, where n is the total number of identical two-ports in the regular substructure. An exact n times speed-up cannot be achieved due to the processing time needed for handling the interconnection of the two-ports, however, the savings in time are still very significant.

Now, suppose that on the next hierarchy level similar savings are obtained, and analysis time of the second level sections is reduced m times. The combined savings in time for the two level hierarchy is proportional to the product of n and m. If the hierarchy is organized so that each level has the same number of sections, for instance n, then the k level hierarchical structure can be analyzed up to n^k times faster than the flat level network. In other words, while a flat level network analysis requires time which is at least proportional to the number of nodes in the regular structure (linear time performance), a hierarchical analysis can be performed in time proportional to the logarithm of the number of nodes. While this performance is based on the theoretical study of the presented method, numerical simulation confirmed this speed-up over the linear algorithms.

4.1. Evaluation of Transfer Functions in Hierarchical Analysis

Transfer functions in hierarchical analysis are evaluated by the algorithm presented in Section 3.2, with the exception that each HOS must first be analyzed to obtain its transmission matrix (Step 1 of the algorithm). This analysis depends on the internal structure of the HOS, the number of hierarchy levels, and the number of identical cells on each level. Regularity of interconnections is used to facilitate this analysis. For instance, if a HOS contains a cascade of identical cells, the analysis presented in the Appendix is executed to obtain its transmission matrix.

The HOS analysis starts at the lowest level in order to obtain transmission matrices of the cells on the higher level. Then, on the higher level, these transmission matrices are used to compute the transmission matrices of still higher level cells, and the process continues until the top level HOS is analyzed. Note that this analysis does not consider the actual input and output terminals of the interconnect analysis and can be executed very efficiently. Steps 2–4 are executed as before, with the transmittance matrices of HOS's used whenever a branch is not a simple structure.

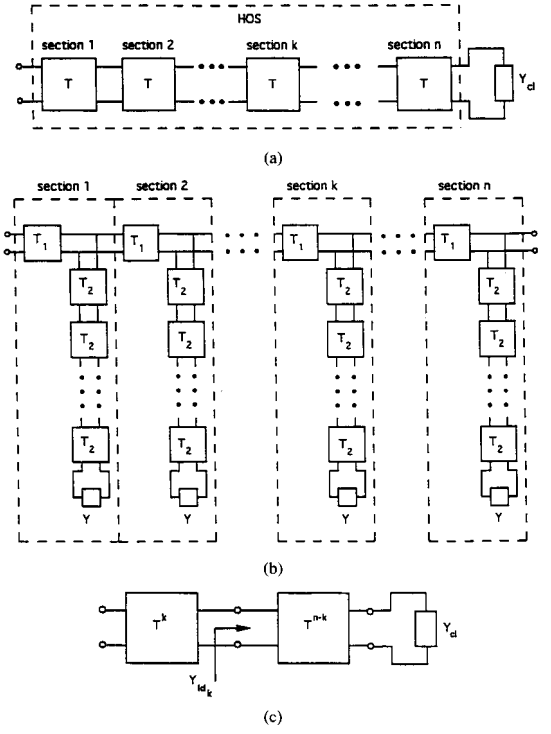

Fig. 2. A hierarchically organized structure (a) cascade of n sections, (b) two level HOS, (c) illustration of the intermediate stage of HOS analysis.

Example 1: As an example of the HOS analysis, consider the two-level HOS shown in Fig. 2(a). It contains a cascade of n identical two-ports, each representing the HOS of the lower level. Each HOS of the lower level contains a two-port with transmission matrix T_1 and a cascade of m identical two-ports with transmission matrices T_2, terminated with a discrete load Y. The expanded two-level HOS is shown in Fig. 2(b). The following values are determined during the HOS analysis:

1) The transmission matrices of the cascades in the lower level HOS's are equal to T_2^m.
2) The transmission matrices of the lower level HOS's are equal to

$$T = T_1 \begin{bmatrix} 1 & 0 \\ Y_{\text{in}} & 1 \end{bmatrix}$$

where $Y_{\text{in}} = (C_m + D_m Y)/(A_m + B_m Y)$ and A_m, B_m, C_m, and D_m are the parameters of the matrix T_2^m.
3) The transmission matrix of the higher level HOS is equal to T^n.

After all HOS's are analyzed, the transfer functions from the input to the individual outputs can be evaluated. If an output is at the output node of the hierarchical branch, or a branch which represents a single two-port, then the transfer functions are evaluated as discussed in Step 5 (Section 3.2). The procedure must be modified only when the output is inside a hierarchical structure. In fact, the only change in the procedure is the evaluation of $T_{i_{\text{out}}}$ (the last matrix in the product computed in Step 5 of the algorithm) and Y_{out} (needed in

Step 6), since these values are not readily available after the HOS analysis step.

First, let us discuss the evaluation of $T_{i_{out}}$. Let us consider a cascade of n identical two-ports and assume that the output is inside the kth two-port (see Fig. 2(a)). Depending on the two-port's internal structure, different ways of evaluating the transfer function from its input to the internal point can be utilized. The most typical structures used in the interconnect networks are Γ and T cells, built of transmission lines or HOS's with discrete loads at their outputs. In order to illustrate the procedure, let us assume that each cell is a Γ section with the horizontal two-port being a simple transmission line, and the vertical two-port being a lower level HOS (similar to Fig. 2(b)).

The transfer function from the input to the output of section k is equal to T^k. At this point the cascade is broken in two parts (see Fig. 2(c)). The first part from the cascade's input to the output of section k (represented by its transmission matrix T^k), and the rest from the output of section k to the output of the cascade. The stray admittance Y_{ld_k} at the output of section k is equal to the input admittance of the rest of the cascade, terminated with the load admittance of the cascade, and can be obtained from

$$Y_{ld_k} = \frac{C_{n-k} + D_{n-k}Y_{cl}}{A_{n-k} + B_{n-k}Y_{cl}}. \tag{5}$$

where $A_{n-k}, B_{n-k}, C_{n-k}$, and D_{n-k} are parameters of the transmission matrix T^{n-k}, and Y_{cl} is the load of the cascade. At the highest hierarchy level, the load admittance Y_{cl} is evaluated during the analysis as the load admittance at the output node (Step 4 of the algorithm). At all the lower levels, this load is equal to the admittance of the discrete load at the end of the cascade.

Having T^k and Y_{ld_k} first use (4) to evaluate T_{ld_k} and then evaluate $T_{i_{out}}$

$$T_{i_{out}} = T^k T_{ld_k} T_{i+1_{out}}. \tag{6}$$

In this equation $T_{i+1_{out}}$ stands for the transmission matrix of the lower level sections, starting from the output of the section k, at the present level, to the desired output (on one or more hierarchy levels below). As we can see, this equation is recursive in nature, and $T_{i+1_{out}}$ can be evaluated in a similar way as $T_{i_{out}}$ until the lowest level is reached. At the lowest level the formula is reduced to

$$T_{i_{out}} = T^k T_{ld_k}. \tag{7}$$

Finally, Y_{out} is evaluated. If the output is defined at the output of the section k at a specific hierarchy level then

$$Y_{out} = Y_{ld_k} \tag{8}$$

where Y_{ld_k} is the load admittance computed from (5) at the given level, otherwise it is the discrete load admittance of the output transmission line.

V. Time Domain Solution

The analyses discussed in Sections III and IV are performed in the complex frequency domain. The time domain response on a specific output is obtained by multiplying the corresponding transfer function by the Laplace transform of the input signal, and finding the inverse Laplace transform. In this section an event driven approach to the inverse Laplace transform is described. Using this approach, the number of explicit time domain evaluations and related frequency domain analyses is reduced without sacrificing the solution accuracy.

5.1. Inverse Laplace Transform

To obtain the output function value in time domain $v(t)$, from its Laplace transform $V(s)$, the numerical Laplace transform inversion technique presented in [18] is used. The desired accuracy of the Laplace inversion is controlled by choosing the order of Pade approximation of e^{st}. In this work, Pade approximation with the denominator degree $M = 8$ and the numerator degree $N = 6$ was selected, which gives 15 correct terms of Taylor expansion of e^{st}. This selection gives sufficient accuracy for interconnect analysis. If a higher accuracy is desired, then more terms of the Taylor expansion of e^{st} must be matched. However, this increases computational cost.

For the selected algorithm, the poles z_i and the residues K_i' from the table presented in [19] are used. Let us suppose that $V(s)$, which is a Laplace transform of $v(t)$, is known. The inverse Laplace transform is computed using

$$v(t) = -\frac{1}{t}\sum_{i=1}^{M'} \mathrm{Re}[K_i' V(z_i/t)] \tag{9}$$

where $M' = M/2$. We see that for the selected accuracy of the inverse Laplace transform, four values of $V(s)$ have to be computed at each time point t at discrete frequencies $s_i = z_i/t$. When a more accurate method, with a larger number of poles, is selected, then the number of complex function evaluations increases. First, each complex frequency value $V(s_i)$ is obtained by performing the complex domain analysis to obtain the transfer function value at a specific frequency as discussed in Sections III and IV. Then, this transfer function value is multiplied by the Laplace transform of the input signal.

5.2. Time Prediction

Fast event driven simulators, like the one presented in [20], evaluate a circuit response only during the time when the voltage in a specified region of the circuit changes by a given value. This approach saves analysis time, since the circuit is not analyzed when its responses do not change or change by a small amount. A similar approach is adopted in this work and is combined with the inverse Laplace transform. In what follows the event driven time prediction is discussed.

Assume that the circuit response at the three most recent time points was evaluated. The next time point, at which a response changes by a specific amount, will be predicted using the quadratic approximation. Note that if the response was changing very slowly at the recent time points, this approximation yields the next time point far away from the most recent one. In general, this reduces the number of time points in which the time domain solution is evaluated. Savings due to this controlled time step depend on the variation of the network responses.

Let us assume that t_{-2}, t_{-1} and t_0 are the most recent time points, and $v(t_{-2}), v(t_{-1})$, and $v(t_0)$ are the corresponding function values. Find a unique parabola, which passes through these three points, and use it to find the closest time point t_1 at which the function value will be $v(t_0) + \Delta v$ or $v(t_0) - \Delta v$, where Δv is a specified voltage deviation. The computational cost for performing this time prediction is that of the solution of a single quadratic equation.

Next, the inverse Laplace transform of the output voltage at t_1 is found. If the obtained value differs from $v(t_0)$ by more than Δv, the time increment is reduced twice and the output function evaluation is repeated. Notice that the accuracy of the solution evaluated at the new time point does not depend on the increment in time. The error depends only on the selected order of the inverse Laplace method and can be reduced to the satisfactory level.

TABLE I
PERFORMANCE COMPARISONS

circuit name	lines	number of elements	nodes	time in sec Connect	Advice
net16_1	16	32	18	8	55
net16_8	16	256	130	8	98
net136_1	136	272	138	12	102
net136_8	136	2176	1090	12	477
net528_1	528	1056	530	5	298
net528_8	528	8448	4226	5	2593

The event driven approach is by itself a significant reduction in analysis time particularly in digital circuits, where signals do not change for longer periods of time. Savings of two to three orders of magnitude for the event driven simulation over the variable step, variable order integration methods were reported [20].

VI. Computer Simulation Results

The complexity of the presented method is linear with the number of different two-ports analyzed. This statement has two important implications on the computing time of the algorithm based on the presented method. First, the number of two-ports is smaller than the number of nodes in the network, particularly if several ladder sections are used to model a transmission line. Other efficient algorithms for the interconnect analysis may also be linear or nearly linear (e.g., the algorithm presented in [7]), but they depend on the number of nodes in the discrete model of the transmission line. Second, if the HOS's are parts of the interconnect network, then the effective number of two-ports for interconnect analysis is reduced to the number of different transmission lines. This reduces the complexity of the method to a logarithmic one and gives savings in time, which may be very significant in the regular VLSI designs.

The interconnect analysis program Connect, which realizes this method, has been written in C++ and implemented on Sun workstations at AT\&T Bell Laboratories. In order to demonstrate the efficiency of the presented method, the analysis time required by the Connect program was compared with the time needed by Advice, a general circuit simulator used in AT\&T Bell Labs. [21]. The results of this comparison are presented in Table I. Both programs were run on Sun-3/60 and results represent combined user time and system time for analysis.

In Table I, the numbers of elements and nodes correspond to discrete models simulated by the Advice program, while the number of lines corresponds to distributed models simulated by the Connect program. The number of nodes in the case of the distributed lines modeled by a single Γ section each, is equal to the number of lines plus 2. The speedup factor is larger for larger structures, for which effectiveness of the hierarchical analysis becomes more evident, and where any overhead related to necessary preprocessing is less significant.

VII. Conclusion

A new method for the interconnect networks analysis has been developed. This method uses exact models of transmission lines and accepts any two-port which can be described by its transmission matrix. The method simulates event driven behavior, since the output is computed only at time instances at which it is expected to change by a specific amount. Numerical inverse Laplace transform is used to obtain time domain solutions. This method works much faster on the interconnect networks than any known timing simulator, and gives accurate results. Savings in time are most significant when the analyzed structures are regular and are described hierarchically. The method has been implemented in C++ on Sun workstations. The computer program Connect, which realizes this method, was tested on several interconnect networks and compared with the circuit simulator Advice. The results of these tests are presented.

VIII. Appendix

Evaluation of the Cascade Connection

Consider a cascade connection of n identical two-ports. Assume that the transmission matrix of a single two-port is equal to

$$T = \begin{bmatrix} A & B \\ C & D \end{bmatrix}. \tag{10}$$

Then the transmission matrix of the entire chain is T^n and T^n can be obtained using eigenvalues and the characteristic equation of the matrix T. Since T is a 2×2 matrix, its eigenvalues can be obtained directly from:

$$\lambda_{1,2} = 0.5\left(A + D \pm \sqrt{(A-D)^2 + 4BC}\right), \tag{11}$$

where A, B, C, and D are the coefficients of the matrix T. In the special case of T being a transmission matrix of a transmission line (1) we get

$$\lambda_{1,2} = \exp(\pm\gamma l). \tag{12}$$

Using the Cayley–Hamilton theorem, nth power of the matrix T can be computed from

$$T^n = \alpha_n T + \beta_n I, \tag{13}$$

where $\alpha_n = (\lambda_1^n - \lambda_2^n)/(\lambda_1 - \lambda_2)$, and $\beta_n = (-\lambda_2\lambda_1^n + \lambda_1\lambda_2^n)/(\lambda_1 - \lambda_2)$. For instance, using (1), (12) and (13) the transmission matrix T^n of a cascade of n identical transmission lines is

$$T^n = \begin{bmatrix} \cosh(n\gamma l) & Z_f \sinh(n\gamma l) \\ \dfrac{\sinh(n\gamma l)}{Z_f} & \cosh(n\gamma l) \end{bmatrix}, \tag{14}$$

which is a well known result for the transmission lines. In a special case, when the two eigenvalues are equal, then $\alpha_n = n\lambda^{n-1}$ and $\beta_n = (-n+1)\lambda^n$.

Acknowledgment

The author wishes to thank Mehdi Hatamian from the VLSI Systems Research Lab. at AT&T Bell Laboratories for his contribution to the development of this method and cooperation during this research. The author would also like to thank Brian Ackland and Robert Clark for helpful discussions during the course of this work.

References

[1] J. E. Buchanan, *BiCMOS/CMOS Systems Design*. New York: McGraw-Hill, 1990.
[2] W. C. Elmore, "The transient response of dumped linear networks with particular regard to wide-band amplifiers," *J. Appl. Phys.*, vol. 19, pp. 55–63, 1948.
[3] J. Rubinstein, P. Penfield, and M. Horowitz, "Signal delay in RC tree networks," *IEEE Trans. Computer-Aided Design*, vol. CAD-2, pp. 202–211, July 1983.
[4] A. Raghunatan and C. D. Thompson, "Signal delays in RC trees with charge sharing and leakage," in *Proc. 19th Assilomar Conf. Circuits, Systems and Computers*, Nov. 1985, pp. 557–561.
[5] C. A. Marinov and P. Neittaanamaki, "A theory of electrical circuits with resistively coupled distributed structures: Delay time predicting," *IEEE Trans. Circuits and Systems*, vol. CAS-35, pp. 173–183, 1988.
[6] L. T. Pillage, "Asymptotic Waveform Evaluation for Timing Analysis," Res. Rep. CMUCAD-89-34, Carnegie-Mellon Univ., 1989.

[7] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design*, vol. 9, pp. 352–366, 1990.

[8] A. H. Zemanian, "Delay time bounds for on-chip and off-chip interconnection networks," in *Proc. IEEE Int. Symp. Circuits and Systems*, Aug. 1990, pp. 2634–2637.

[9] P. R. O'Brian and T. L. Savarino, "Modeling of the driving-point characteristic of resistive interconnect for accurate delay estimation," in *Proc. IEEE Int. Conf. Comp. Aided Design*, Nov. 1989, pp. 512–515.

[10] S. P. McCormick and J. Allen, "Waveform moment methods for improved interconnection analysis," in *Proc. 27th ACM/IEEE Design Autom. Conf.*, June 1990, pp. 406–412.

[11] D. H. Xie and M. Nakhla, "Delay and crosstalk simulation of high-speed VLSI interconnects with nonlinear terminations," in *Proc. IEEE Int. Conf. Comp. Aided Design*, Nov. 1991, pp. 66–69.

[12] T. Sakurai, "Approximation of wiring delay in MOSFET LSI," *IEEE J. Solid State Circuits*, vol. SC-18, pp. 418–426, Aug. 1983.

[13] N. Gopal, D. P. Neikirk, and L. T. Pillage, "Evaluating RC-interconnect using moment-matching approximations," in *Proc. IEEE Int. Conf. Comp. Aided Design*, Nov. 1991, pp. 74–77.

[14] C. Chu and M. Horowitz, "Charge sharing models for switch-level simulation," *IEEE Trans. Computer-Aided Design*, vol. CAD-6, pp. 1053–1060, 1987.

[15] J. S. Roychowdhury, A. R. Newton, and D. O. Pederson, "An impulse-response based linear time-complexity algorithm for lossy interconnect simulation," *Proc. IEEE Int. Conf. Comp. Aided Design*, Nov. 1991, pp. 62–65.

[16] D. Winklestein, M. B. Steer, and R. Pomerleau, "Simulation of arbitrary transmission line network with nonlinear terminations," *IEEE Trans. Circuits and Syst.*, vol. 38, pp. 418–421, 1991.

[17] S. Lin and E. S. Kuh, "Pade approximation applied to lossy transmission line circuit simulation," in *Proc. IEEE Int. Symp. Circuits and Syst.*, May 1992, pp. 93–96.

[18] J. Vlach and K. Singhal, *Computer Methods for Circuit Analysis and Design*. Amsterdam, The Netherlands: Van Nostrand Reinhold Co., 1983.

[19] K. Singhal, J. Vlach, and M. Nakhla, "Absolute stable, high order method for time domain solution of networks," *Archiv fur Electronik und Ubertraugungstechnik* (Electronics and Communication), vol. 30, pp. 157–166, 1976.

[20] B. D. Ackland and R. A. Clark, "Event-EMU: An event driven timing simulator for MOS VLSI circuits," in *Proc. IEEE Int. Conf. Comp. Aided Design*, Nov. 1989, pp. 80–83.

[21] T. A. Pfannkoch, "Advice 1N User's Guide," TM 52173, AT&T Bell Laboratories, Apr. 1988.