High Speed Digital System Simulation using Frequency Dependent Transmission Line Network Modeling

Mark S. Basel[1], Michael B. Steer[1], Paul D. Franzon[1] and Dan Winkelstein[2]

[1] North Carolina State University
[2]Microelectronics Center of North Carolina

Abstract

A robust and accurate method for the analysis of high speed digital circuits with a lossy, frequency-dependent transmission line network is presented. Implementation in an analog circuit simulator, using convolution and time-domain scattering parameters, in combination with a linear microwave circuit simulator is discussed.

I. Introduction

The high speed performance of most digital systems is now limited by distributed discontinuities and transmission line effects of packaging and interconnects, and not by the switching speed of semiconductor devices. Among the effects of these distributed circuit elements are: Delay and rise time degradation, ringing, crosstalk noise, metastates and ground bounce.

In many cases, the quest for higher performance is centering on layout and topological considerations, not just increasing device clocking rates.

The fundamental difficulty encountered in incorporating transmission line simulation in a transient circuit simulator arises because circuits containing nonlinear devices or time dependent characteristics must be characterized in the time domain while transmission lines networks, which have frequency dependent effects such as loss, dispersion, and interconnect discontinuities are best simulated in the frequency domain.

Several approaches to this problem have been tried with varying degrees of success. The Laplace Transform technique is good at handling multiconductor transmission line systems and while it is more efficient than the convolution technique, its limited ability to handle frequency dependent effects and problems dealing with nonlinear terminations restrict its' usefulness. The Method of Characteristics [9] is relatively easy to use with conventional circuit simulators, it's restricted to low loss, dispersionless lines (i.e. with frequency independent parameters). Hybrid schemes that use harmonic balance to handle the distributed elements in the frequency domain and time-domain nonlinear methods for dealing with the discrete components have also been used [11]. Other techniques for simulating distributed element systems have recently been reviewed by Djordjevic et al [1, 2] and Schutt-Aine and Mittra [3, 4].

In one of Djordjevic et al techniques the frequency domain Y parameter description of the distributed network is converted to a time-domain description using a Fourier transform [5]. While this technique can handle lossy coupled networks, a difficulty arises as the Y parameters of a typical multiconductor array have a wide dynamic range. Consequently aliasing in the frequency-domain to time-domain transformation can cause appreciable errors in the simulated transient response.

An alternative formulation that avoids these problems is that of Schutt-Aine and Mittra [4] who use a scattering parameter formulation to consider a parallel coupled transmission line system. Because of the limited range of scattering parameters (i.e. 0 to 1), this approach offers good computational stability and efficiency.

Time-domain scattering parameters have also been used to model transmission lines in sub-nanosecond GaAs MESFET integrated circuits, but only for uncoupled lines [8].

We have previously presented [10] another more direct S parameter technique based on the one of the approaches of Djordjevic et al [5]. We derive a Green's function from the frequency-domain S parameters and use this to develop a method for analyzing an arbitrarily complex transmission line network terminated in nonlinear loads. For the reasons given above, this technique has reduced sensitivity to dynamic range problems than approaches based on network parameters other than S parameters. Nonuniform lines can be considered and the transmission line networks can be defined by network parameters (e.g. Y or S parameters) derived through measurement or circuit simulation. Previously [10], implementation in a transient circuit simulator was discussed. Here we present an extension of the method to a generalized packaging analysis transient simulator.

II. Convolution Method

S parameters of a transmission line system describe the relative amplitude and phase of the forward and backward traveling waves at each port and at each frequency on a transmission line of characteristic impedance Z_m. With V_j^+ and V_j^- the forward and backward traveling waves respectively at the j th port, a frequency-domain S parameter is just the ratio of the backward traveling wave to the forward traveling wave, $S_{ij}(\omega) = V_i^-(\omega)/V_j^+(\omega)$. Thus each S parameter has a direct physical relationship with a reflection or coupling process. Each time-domain S parameter (being the Fourier transform of the corresponding frequency-domain S parameter) is then the response of a backward traveling wave with respect to an impulse forward traveling wave at one of the ports of the network with all ports terminated in the reference impedance, Z_m, of the S parameters, i.e. $S_{ij}(t) = V_i^-(t)/V_j^+(0)$.

Integration into a transient circuit simulator requires that the description of the transmission line system be in terms of total voltages rather than traveling wave components. This can be achieved by terminating each port in its reference impedance so ensuring that there are no reflections at the ports of the distributed network as shown in Fig. 1. The total transient response at port i of a transmission line system to a total voltage $E_j(t)$ $(= V_j^+(t) + V_j^-(t))$ with output impedance Z_m at each port is[10]

$$V_i(t) = \sum_{j=1}^{N} \int_{-\infty}^{t} g_{ij}(t-\tau)E_j(\tau)d\tau = \sum_{j=1}^{N} g_{ij}(t) \star E_j \quad (1)$$

Here N is the number of external ports in the transmission line system and $g_{ij}(t)$ is the Inverse Fourier transform of $G_{ij}(\omega)$.

Removing the effect of the reference impedance (see [10]) leads to a set of convolution equations which have the discrete form

$$\begin{aligned} V_i(n_t) &= \sum_{j=1}^{N} \left[\sum_{n_\tau = 0}^{n_t} g_{ij}(n_t - n_\tau) V_j'(n_\tau) \right] \\ &+ \sum_{j=1}^{N} \left[\sum_{n_\tau = n_t + 1}^{N_T} g_{ij}(n_\tau) V_j'(0) \right] \end{aligned} \quad (2)$$

$$V_j'(n_t) = V_j(n_t) - I_j(n_t) Z_m \qquad j = 1, N \quad (3)$$

In (2) N_T is the number of time points in the period of interest, $t = \Delta t . n_t$, and $\tau = \Delta\tau . n_\tau$. The introduced node sources V_i are now memory devices controlled by the present and past value of all virtual sources, the V_j''s, according to (2).

Equations (3) and (2) are a set of coupled nonlinear equations and must be solved iteratively. In iterative vector form (3) becomes

$$^{k+1}\hat{V}'(n_t) = {}^{k}\hat{V}(n_t) - {}^{k}\hat{I}(n_t) Z_m \quad (4)$$

and (2) becomes

$$^{k+1}\hat{V}''(n_t) = \mathbf{\Lambda}\, {}^{k+1}\hat{V}'(n_t) + \hat{\alpha}(n_t) \quad (5)$$

where $\hat{\alpha}(n_t)$ is a vector with elements

$$\alpha_i = \sum_{j=1}^{N} \left[\sum_{n_\tau = 0}^{n_t - 1} g_{ij}(n_t - n_\tau)\, V_j'(n_\tau) + \sum_{n_\tau = n_t + 1}^{N_T} g_{ij}(n_\tau)\, V_j'(0) \right] \quad (6)$$

$\mathbf{\Lambda}$ is a matrix with elements $\lambda_{ij} = g_{ij}(0)$. and $^{k+1}\hat{V}(n_t)$ is chosen to minimize $|{}^{k}\hat{V}(n_t) - {}^{k+1}\hat{V}''(n_t)|$. Then

$$^{k+1}\hat{V}(n_t) = {}^{k}\hat{V}(n_t) - \mathbf{J}^{-1}\left({}^{k}\hat{V}(n_t) - {}^{k+1}\hat{V}''(n_t)\right) \quad (7)$$

Iteration would then proceed until some predetermined tolerance is obtained. That is until $|{}^{k+1}\hat{V}''(n_t) - {}^{k}\hat{V}(n_t)| < \epsilon$.

The extension to this method used in the new simulator comes about if the terminations are considered to be instantaneously uncoupled. The above simplifies considerably so that, $g_{ij}(0) = 0$ for $i \neq j$, so that $\mathbf{\Lambda}$ is now a diagonal matrix with elements $\lambda_{ii} = g_{ii}(0)$ so that (5) becomes

$$^{k+1}V_i''(n_t) = g_{ii}(0)\left[{}^{k}V_i(n_t) - {}^{k}I_i(n_t)Z_m\right] + \alpha_i(n_t) \quad (8)$$

Thus, (7) reduces to

$$^{k+1}V_i(n_t) = {}^{k}V_i(n_t) - \gamma_{ii}\left({}^{k}V_i(n_t) - {}^{k+1}V_i''(n_t)\right) \quad (9)$$

with

$$\gamma_{ii} = \left(1 - g_{ii}(0)\left[1 - Z_m \frac{\partial\, {}^{k}I_i(n_t)}{\partial\, {}^{k}V_i(n_t)}\right]\right)^{-1} \quad (10)$$

III. Example

The above equations are considerably simplified in the case of a two port. Equation (4) becomes

$$^{k+1}V_1'(n_t) = {}^{k}V_1(n_t) + {}^{k}I_1(n_t)Z_m \quad (11)$$

and

$$^{k+1}V_2'(n_t) = {}^{k}V_2(n_t) + {}^{k}I_2(n_t)Z_m \quad (12)$$

Equation (5) becomes

$$^{k+1}V_1''(n_t) = g_{11}(0)\, {}^{k+1}V_1'(n_t) + \alpha_1 \quad (13)$$

and

$$^{k+1}V_2''(n_t) = g_{22}(0)\, {}^{k+1}V_2'(n_t) + \alpha_2 \quad (14)$$

The Newton iteration scheme becomes

$$\begin{aligned} ^{k+1}V_1(n_t) = \; & {}^{k}V_1(n_t) - \left[1 - g_{11}(0)\left(1 - Z_m \frac{\partial\, {}^{k}I_1(n_t)}{\partial\, {}^{k}V_1(n_t)}\right)\right]^{-1} \\ & \times \left({}^{k}V_1(n_t) - {}^{k+1}V_1''(n_t)\right) \end{aligned}$$

and

$$\begin{aligned} ^{k+1}V_2(n_t) = \; & {}^{k}V_2(n_t) - \left[1 - g_{22}(0)\left(1 - Z_m \frac{\partial\, {}^{k}I_2(n_t)}{\partial\, {}^{k}V_2(n_t)}\right)\right]^{-1} \\ & \times \left({}^{k}V_2(n_t) - {}^{k+1}V_2''(n_t)\right) \end{aligned}$$

IV. Discussion of TRANSIM

The above algorithm for transmission line network analysis has been implemented in a transient circuit simulator (known as TRANSIM, for TRANsient SIMulator) which allows user defined nonlinear controlled voltage sources as well as subcircuit or macros.

The circuit to be analyzed is partitioned into two subcircuits, the interconnection network and the nonlinear loads (devices). In most physical layouts, relatively few different interconnection structures are used since most autorouters are restricted to a specific set of geometries for elements like bends and vias. Transmission lines also have restrictions as to widths, spacings, etc. Since most routers try to maximize density, structures such as data busses usually have the closest allowable spacing and vary little in appearance but for their length.

In the simulator, the network of interconnection lines (including junctions such as vias, bends, etc.) are treated as linear circuit elements. Models for each subassembly (such as a bend) are evaluated and the admittance matrix parameters determined. For any particular structure, this operation is only performed once. On each successive encounter with the same element type, pointers are assigned to the previously and stored calculated data; thus eliminating unnecessary computations. Once all of the individual interconnection elements have been calculated, a nodal admittance matrix is produced which incorporates all of the Y parameter matrices of the individual elements.

This resulting large matrix could be used in the transient analysis of the complete circuit but the computing time would be correspondingly large. Since circuit designers are mostly interested in what is going on at the device terminals, the voltages and currents at individual nodes throughout the interconnection network are of little interest. The simulator can be 'forced' to keep track of internal nodes by declaring them as 'outside' terminals.

By performing a matrix reduction on the complete nodal admittance matrix, the size of the problem can be reduced to a $M \times M$ admittance matrix where M is the number of nonlinear loads (e.g. the I/O's on the IC's) which the interconnection network joins together.

Further speedups can be initiated by the user if (for noncritical areas) the nonlinear loads are replaced with linear elements. These loads can then be incorporated into the complete nodal admittance matrix and later reduced down to an even smaller matrix.

Once the nodal admittance matrix has been reduce, the Y parameters are still in the frequency dependent form. Following the proceeding development of the algorithm, the admittance matrix is transformed into and S parameter matrix and an FFT is performed on the resulting matrix to produce a time domain Green's function matrix. The transient analysis is performed according to the description of the algorithm.

The simulator can be broken down to four basic sections (as shown in Fig. 2); the CORE, Physical Interface, Network Level and Output Functions. The CORE is a small section of code with the primary function of setting up the initial structures and calling the other sections. At present, the CORE setups up the structures according to a netlist interpreted by a parser (a graphical interface is a future addition). As each element is read in, CORE sets up the storage requirements for each element by calls through the Physical Interface. CORE doesn't know anything about the elements themselves, that is left to the Physical Interface to handle. Thus TRANSIM is a model independent simulation and new elements can be added in a matter of hours. At this time, CORE also determines which terminals are outside (i.e. nonlinear loads) terminals and keeps track of this information for the Network Level. Each unique element has a data structure associated with it. Within this structure are locations for the physical parameters associated with the element, Y parameter data (generated later) and pointers to the routines that do the actual calculations of the Y parameters.

After the initial setup, CORE then traverses through the elements and calls the evaluation function associated with each. The data so generated is stored in the Element Structure. If the CORE determines that data for a particular element (with the same physical parameters) has been generated already, a pointer is assigned to that previously produced data, avoiding unnecessary reevaluations. The functions used to calculate the Y parameter data can easily be modified or replaced by the user.

Once all the Element Data has been produced, CORE then calls the Network Level routines which produce the complete nodal admittance matrix. CORE then using the Network Level routines along with information about which terminals are 'outside' to determine the reduced nodal admittance matrix.

The transient analysis is then performed and the results interpreted by the Output Functions; raw data files are produced and/or the associated X-Windows plotting package is called to display the results.

V. Results and Discussion

Verification of the transmission line model and its implementation was undertaken for the transmission line system of Fig. 3. The transient response predicted by out model is compared with experimental results in Figs. 4 and 5. The digital devices were modeled at the transistor level using models supplied by the manufacturer.

The S parameter Green's function method developed here was also compared to Djordjevic et al's Y parameter Green's function method [5] for a pair of coupled lines with nonlinear terminations. In this case the two methods yielded very similar transient results. If Z_C were not known, the time-domain Y parameters would extend indefinitely for a low loss line. Since we use S parameters in the development of our Green's function, the length of the impulse response can be conveniently limited in time by low pass filtering of the S parameters.

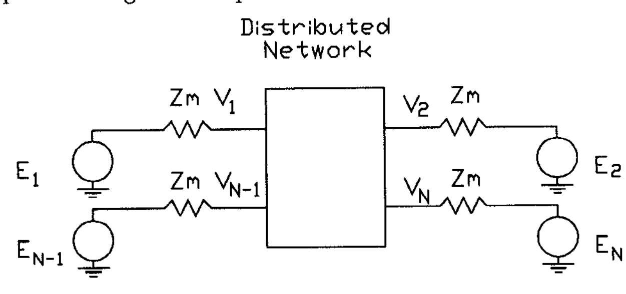

Figure 1: N port distributed network with reference impedance termination and arbitrary sources

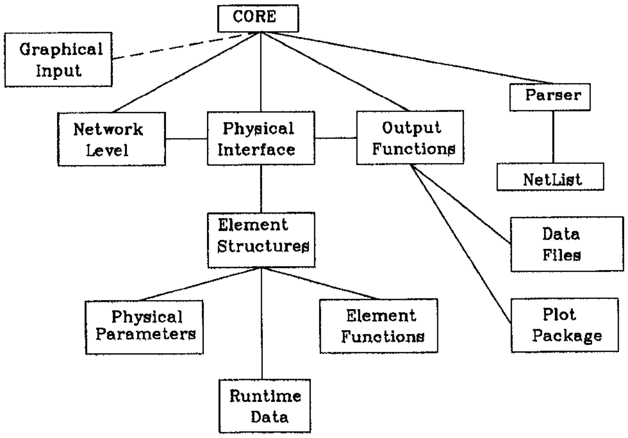

Figure 2: Block Diagram of TRANsient SIMulator

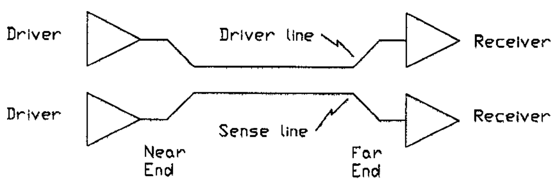

Figure 3: Test Structure

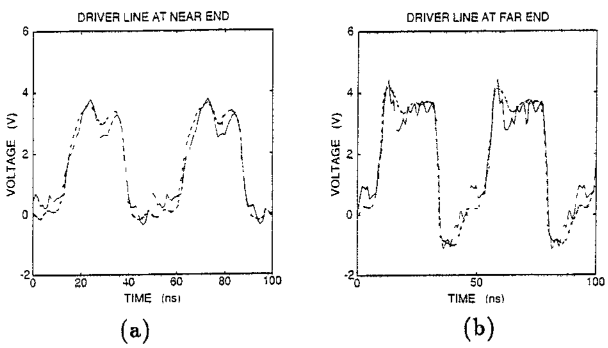

Figure 4: Near (a) and far (b) ends of the driver line (dotted = measured, solid = simulation)

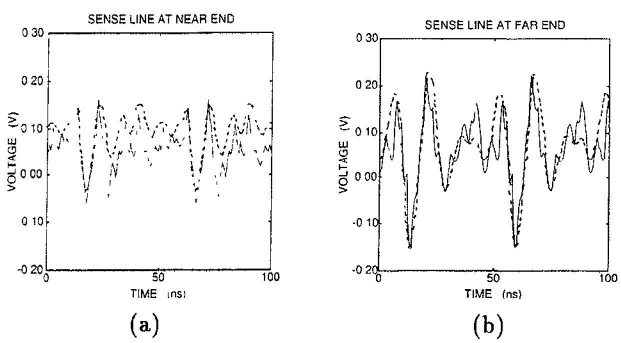

Figure 5: Near (a) and far (b) ends of the sense line (dotted = measured, solid = simulation)

ACKNOWLEDGEMENT

This work was supported by a National Science Foundation Presidential Young Investigator Award Grant No. ECS-8657836.

REFERENCES

[1] A.R. Djordjevic, T.K. Sarkar and Roger F. Harrington, "Time-domain response of multiconductor transmission lines," Proc. I.E.E.E., **75**, pp. 743-764, June 1987.

[2] A.R. Djordjevic and T.K. Sarkar, "Analysis of time domain response of lossy multiconductor transmission line networks," I.E.E.E. Trans. Microwave Theory Tech., **MTT-35**, pp. 898-908, Oct. 1987.

[3] J.E. Schutt-Aine and R. Mittra, "Scattering parameter transient analysis of transmission lines loaded with nonlinear termination," I.E.E.E. Trans. on Microwave Theory and Techniques, **MTT-36**, pp. 529-536, March 1988.

[4] J.E. Schutt-aine and R. Mittra, "Nonlinear transient analysis of coupled transmission lines," *IEEE Trans. Circuits and Systems*, Vol. CAS-36, pp. 959-967, July 1989.

[5] A.R. Djordjevic, T.K. Sarkar, and R.F. Harrington, "Analysis of lossy transmission lines with arbitrary nonlinear terminal networks," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-34, June 1986, pp. 660-665.

[6] R.E. Collin, *Foundations of Microwave Engineering*, McGraw-Hill, 1966, section 4.7

[7] FAST Data Manual, Signetics Corporation, 1987.

[8] M. Passlack, M. Uhle, H. Elschner, "Analysis of Propagation Delays in High-Speed VLSI Circuits using a distributed line model,"*IEEE Trans. Computer-aided design*, vol. 9 no. 8, August 1990, pp.821-826.

[9] F.Y. Chang, "The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines," *IEEE MTT-S Digest*, 1989, pp. 653-656.

[10] D. Winkelstein, M.B. Steer and R. Pomerleau, "Simulation of the Transient Response of Abitrary Transmission Line Networks with Nonlinear Terminations," *IEEE Trans. on Circuits and Systems*, April 1991.

[11] EEsof, Westlake Village, California