EFFICIENT MOMENT-BASED TIMING ANALYSIS FOR VARIABLE ACCURACY SWITCH LEVEL SIMULATION

Russell Kao
Mark Horowitz

Technical Report No. CSL-TR-91-468

April 1991

This research has been supported by DARPA contract N00014-87-K-0828.
Authors also acknowledge support from Digital Equipment Corporation.

EFFICIENT MOMENT-BASED TIMING ANALYSIS FOR VARIABLE ACCURACY SWITCH LEVEL SIMULATION

Russell Kao and Mark Horowitz

Technical Report: CSL-TR-91-468

April 1991

Computer Systems Laboratory
Departments of Electrical Engineering and Computer Science
Stanford University
Stanford, California 94305-4055

Abstract

We describe a timing analysis algorithm which can achieve the efficiency of RC tree analysis while retaining much of the generality of Asymptotic Waveform Estimation. RC tree analysis from switch level simulation is generalized to handle piecewise linear transistor models, non tree topologies, floating capacitors, and feedback. For simple switch level models the complexity is O(n). The algorithm allows the user to trade off efficiency vs accuracy through the selection of transistor models of varying complexity.

Kev Words and Phrases: Asymptotic Waveform Estimation, Moment analysis, RC tree analysis, switch level simulation

Efficient Moment-Based Timing Analysis for Variable Accuracy Switch Level Simulation

Russell Kao and Mark Horowitz

April 8, 1991

Abstract

We describe a timing analysis algorithm which can achieve the efficiency of ***RC** tree analysis* while retaining much of the generality of *AWE*. RC tree analysis from switch level simulation is generalized to handle piecewise linear transistor models, non tree topologies, and feedback. For simple switch level models the complexity is $O(n)$. The algorithm is appropriate for variable accuracy switch level simulation.

Topics of Interest: 1.3, 1.1

1 Introduction

Switch level simulators take advantage of the fact that the full generality of a circuit simulator is unnecessary for predicting the first order behavior of most digital MOS circuits. In order to achieve increased simulation speed, switch level simulators [Ter83] decompose the circuit into small clusters, restrict the topology of clusters to transistor-capacitor trees, and utilize a switched resistor transistor model. ***RC*** tree analysis and the Elmore delay [Elm48, RPH83, Hor83] are used to estimate waveforms and delays. Because of these simplifications the analysis can be very fast; up to three orders of magnitude faster than circuit simulation. Because the computational complexity of switch level algorithms is $O(n)$ [Hor83], *these simulators can* be applied to entire integrated circuits. Frequently, however, small portions of the circuit must be simulated at the circuit level because the simplifications do not allow the accurate prediction of their behavior.

Recently, Pillage and Rohrer[PR90] invented ***Asymptotic Waveform Evaluation (AWE),*** which essentially extends the Elmore Delay to allow arbitrarily accurate estimates of the response of any linear circuit. Unfortunately, the ***Tree-Link Analysis*** used by AWEsim for the computation of moments is not as efficient as ***RC*** tree ***analysis*** for the particular case of RC trees. The complexity of Tree-Link Analysis applied to RC trees is $O(n^2)$ [PD90] rather than $O(n)$. Furthermore, RC tree analysis has been generalized [Chu88] to handle multiple sources while retaining linear complexity. Tree-Link Analysis is $0(n^3)$ for this class of circuits.

Our approach is to compute moments using an extension of RC tree analysis. We extend it along two dimensions. First, we show that the tree analysis can be applied even when transistor models are generalized from resistors to arbitrary piecewise linear devices. For any particular circuit state the complexity remains $O(n)$. Second, we show that ***tearing*** can be used to handle non-tree topologies and feedback. If the number of branches which need to be tom in order to get a feedback-free tree is small and bounded, then even these more complex circuits can be analyzed efficiently. Thus we have a timing analysis which is efficient for simple device models and topologies but which retains much of the generality of AWE. It can be used for estimating cluster delays for a variable accuracy switch level simulator. We expect its speed to be comparable to switch level simulation for the simple switch level models required by most circuits in a digital MOS IC. However the simulator should be able to simultaneously simulate the more complex circuits with greater accuracy at the cost of reduced efficiency for just those circuits.

2 Clusters in Switch Level Simulation

Switch level simulators partition the circuit into channel connected networks referred to as clusters [Ter83]. MOS gates constitute the inputs to a cluster while the outputs are nodes which drive other gates. It is assumed that inputs and outputs are unidirectional (a transistor's gate affects its source and drain but not vice versa) and are at most loosely coupled (there is no feedback). These assumptions allow clusters to be analyzed independently. The interactions between clusters are managed using a selective trace algorithm.

A cluster's response to an input transition is estimated using ***moment analysis.*** Moment analysis is a two stage process involving, first, the computation of the moments of all circuit variables and second, the generation of waveform and delay estimates from those moments[Hor83, PR90]. Moment computation is potentially the most problematic because its complexity is a function of the cluster size.

3 Moment Computation

Moments [1] are computed through the repeated DC solution of a linear network[Chu88, PR90]. To illustrate the mechanics of the procedure, consider the RLC network in Figure 1. Let M_{C_k} and M_{L_k} stand for the kth moments of the capacitor voltage and inductor current, respectively. The lowest order moments, $M_{C_{-1}}$ and $M_{L_{-1}}$, are defined to be the initial capacitor voltage and inductor current. In general the $k+1$th moments are computed from the kth moments by replacing the capacitor with a current source of CM_{C_k} amps and the inductor by a voltage source of LM_{L_k} volts and finding the DC solution of the resulting network. Once the DC solution is

[1] Moments are essentially the coefficients of the series expansion in s of the Laplace transform of the homogeneous portion of the response.

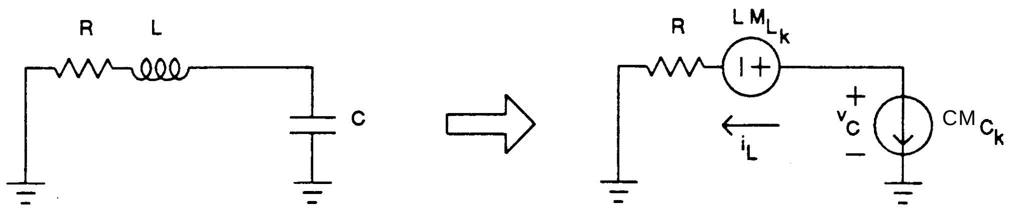

Figure 1: Transform circuit for moment computation.

known $M_{C_{k+1}}$ is given by the voltage across the current source and $M_{L_{k+1}}$ by the current through the voltage source.

Thus, a crucial step in the implementation of any moment analysis is the efficient DC solution of a linear network. In their work on AWE, Pillage, Rohrer et. al. proposed the use of, first, Tree-Link Analysis [PR90] and, later, Path Tracing [PD90]. The advantage of these methods is that they are general and can be applied to any linear network However, both methods require the creation of the Loop/Cutset matrix For the particular case of RC trees the Loop/Cutset matrix has $0(n^2)$ entries. Thus the complexity of these algorithms must be at least $0(n^2)$[2].

4 DC Solution of Leaky Resistor Trees

The earliest switch level simulators [Ter83] modeled MOS transistors using resistors and restricted the circuit topology to RC trees to allow efficient ($0(n)$) solutions. It was observed that the topologies of most digital MOS circuits were, in fact, trees driven by a single voltage source. Later, Chu [Chu88] extended this analysis to include RC trees driven by multiple sources (Figure 2). These topologies may

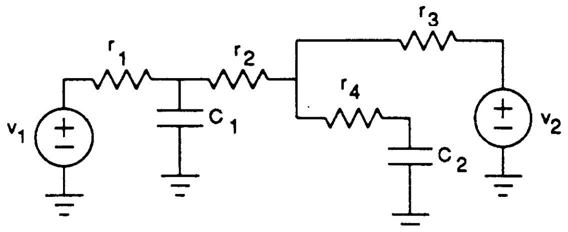

Figure 2: Leaky resistor tree.

continue to be viewed as trees if one redefines leaf nodes to be nodes connected to one transistor terminal and either a grounded current or voltage source[3]. The root

[2]However, Path-Tracing does reduce the number of the most expensive operation, multiply, to 0 (n).
[3]We include nodes connected to current sources with zero current.

node is arbitrarily selected from the non-leaf nodes (See Figure 3). We will refer to such topologies as a ***leaky trees.***

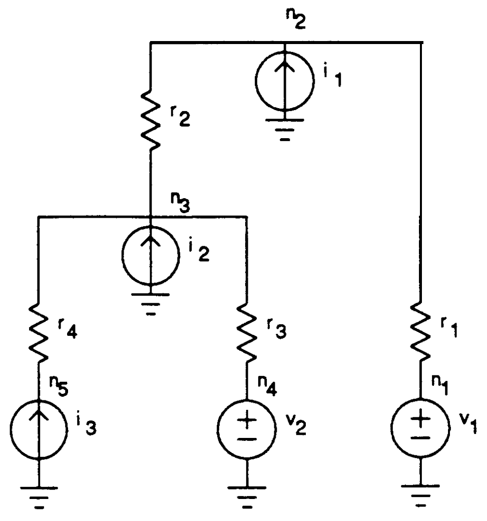

Figure 3: Computing moments for leaky tree.

To review Chu's approach[4], consider the leaky tree in Figure 3 which results when the capacitors are replaced by current sources for the purpose of computing moments. The circuit has been redrawn to suggest its tree structure, with the leaf nodes n_1, n_4, and n_5 at the bottom and the root node n_2 at the top. The DC solution can be found by making two passes over the network. The first pass starts at the leaves of the tree and ascends to the root. The second pass starts at the root of the tree and descends to the leaves.

We begin the first pass with the resistors connected to leaf nodes (r_1, r_3, and r_4). For each resistor we compute the Norton equivalent seen looking into its upper terminal. Once we have computed the Norton equivalents of all resistors descending from a particular node (after the first iteration n_3 becomes such a node) we can combine them with the capacitor current sources to produce the Norton equivalent seen looking out of the lower terminal of the (single) resistor ascending from that node (for n_3 this is r_2). We record this aggregate Norton equivalent at the node for use in the second pass. The first pass continues iteratively, replacing each ascending resistor by the Norton equivalent seen looking into its upper terminal and, in turn, computing the Norton equivalent seen by the parent node's ascending resistor. The iteration terminates when we have computed the Norton equivalent of all the resistors descending from the root node.

[4]Chu's thesis describes the analysis in terms of "moving capacitors". We present his work from the slightly different perspective of Norton analysis applied to the network.

The second pass starts by solving for the voltage at the root node. This is possible because the root node has no resistors ascending from it and in the first pass we saved for each node the Norton equivalent of all resistors (and capacitor current sources) descending from that node. Once we know the root's voltage, we can solve for the voltage of its children by utilizing the Norton equivalent saved at each child. We continue descending the tree until we've solved for all the voltages.

In summary, the process utilizes two mapping computations. In the first pass (See Figure 4) we are given g_1, i_1, and g and need to find g_2 and i_2

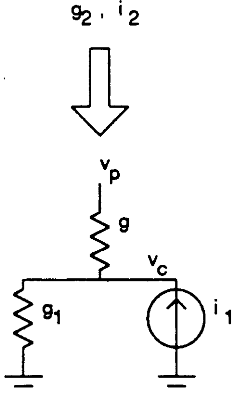

Figure 4: Norton calculations.

$$g_2 = \frac{g_1 g}{g_1 + g} \tag{1}$$

$$i_2 = i_1 \frac{g}{g_1 + g} \tag{2}$$

In the second pass we are given, in addition, v_p and we need to find v_c

$$v_c = \frac{v_p g + i_1}{g_1 + g} \tag{3}$$

Later, we will show that these two computations can still be performed even if resistor g is replaced by a transistor modeled using piecewise linear functions.

5 PWL Transistor Models

Switch level simulators simulate the behavior of digital MOS circuits by replacing the transistors with resistors[Ter83]. Experience with these simulators has shown that the simple resistor model suffices to predict the behavior of most digital MOS circuits.

However, it is not uncommon for a large chip to have a small number of circuits whose behavior must be verified using a more accurate simulator such as SPICE. Furthermore there has been an increasing interest in the incorporation of bipolar transistors into digital integrated circuits. The behavior of bipolar transistors, for example in ECL circuits[KAHS88], cannot be adequately modeled using switched resistors. In order to allow for more sophisticated transistor models we employ piecewise linear models. In this context, the switched resistor is viewed as a simple 2–region piecewise linear device.

Piecewise linear models have been used by a number of simulators employing numerical integration[vB87, HPR87]. A piecewise linear function of n variables[vB87] can be thought of as a list of linear[5] functions, each of which has associated with it a list of one or more linear inequalities which define the polyhedral region in $\Re^n$ in which that function applies. For example, the current through a simple piecewise linear diode may be modeled by

$$i_{diode} = \begin{cases} 0 & \\ (v_+ - v_-)/r_{diode} & 0 \leq (v_+ - v_- - v_{on}) \end{cases} \quad (4)$$

We will examine interconnections of three terminal piecewise linear devices with the initial restriction that the third terminal is connected to a voltage source. It is, perhaps, surprising that many of the network analysis techniques which have been developed for resistors apply to networks of these devices as well.

5.1 Leaky Trees of Three Terminal Networks

In order to compute the moments of a transistor-capacitor tree we need to find the DC solution of a corresponding tree obtained by setting independent DC sources to zero, replacing inductors with voltage sources, and replacing capacitors with current sources. Furthermore, because we assume inputs are unidirectional and clusters can be analyzed independently, MOS gates are considered to be driven by independent, possibly exponentially time varying voltage sources. It can be shown that when formulating the circuit to compute the $k+1$th moments, each time varying source should be replaced by a DC source set equal to the $k+1$th moment of its waveform (see Figure 5).

From the definition of a piecewise linear function we can see that for any particular polyhedral region each piecewise linear device is equivalent to a linear network. Assuming that each transistor remains in its present state, we group each transistor with the voltage source driving its gate and represent the interface that the pair presents to ***the*** network by the ***short-circuit admittance parameters*** of a ***three terminal network***[BS65][6] (Figure 6).

[5] more accurately, affine

[6] Note that this approach is taken by *macro modeling* [HSV81]

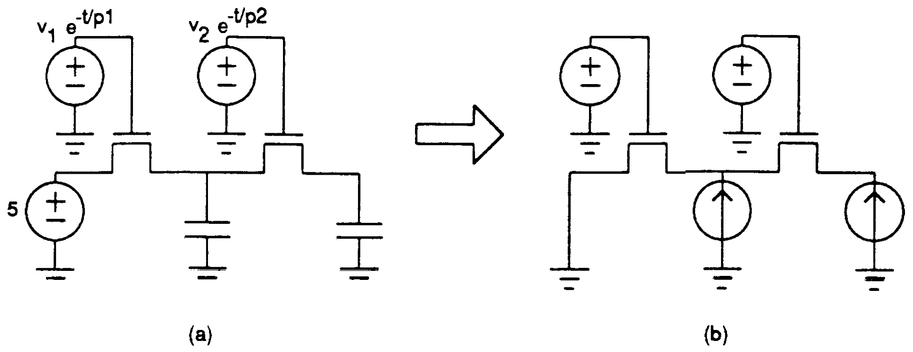

Figure 5: Transistor-capacitor tree.

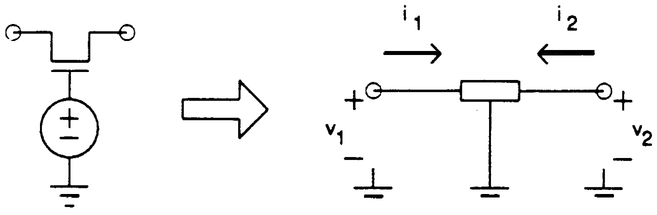

Figure 6: Three terminal network model.

The parameters are defined by extracting two voltage ports, one from each terminal to ground[CL75][7].

$$\begin{bmatrix} i_1 \\ i_2 \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} \\ y_{21} & y_{22} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} + \begin{bmatrix} i_{s1} \\ i_{s2} \end{bmatrix} \tag{5}$$

Figure 7 gives a physical interpretation of the six parameters of the admittance formulation.

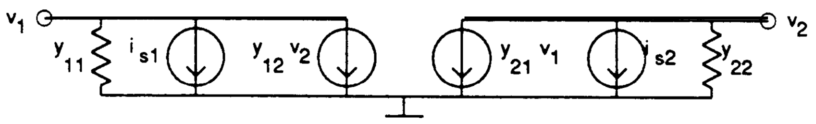

Figure 7: Circuit interpretation of admittance parameters.

In order to find the DC solution of leaky trees of these networks, we need to be

[7]Note that it is not always possible to extract two voltage ports for a particular model. For example admittance parameters cannot be determined for a voltage source. Such devices must be handled as special cases. However, to simplify the discussion we will assume that the admittance representation exists.

able to do two things. If we replace resistor $_g$ in Figure 4 by the circuit in Figure 7 it can be shown that if g_1 and i_1 are known then g_2 and i_2 are given by

$$i_2 = i_{s1} + \frac{y_{12}}{y_{22} + g_1}(i_1 - i_{s2}) \tag{6}$$

$$g_2 = y_{11} - \frac{y_{12} y_{21}}{\mathrm{Y22} + \mathrm{Yl}} \tag{7}$$

If, in addition, v_p is known then v_c is given by

$$v_c = \frac{g_1 + y_{22}}{g_1 y_{22}}(i_1 - i_{s2} - y_{21} v_p) \tag{8}$$

Thus it is possible to generalize the DC analysis of leaky trees of resistors to leaky trees of three terminal networks. Because the above equations take a constant amount of time to compute, the leaky tree analysis remains $O(n)$ in the number of networks in the circuit irrespective of the complexity of the models.

5.2 Series-Parallel Combination

The analogy **with** resistors goes even further. These three terminal networks are also amenable to series-parallel combination. The admittance parameters of the parallel combination of two networks can be found by simply sumrning their corresponding parameters. The parameters of a series combination can be derived from the series combination of two of the circuits in Figure 7. If we let superscripts of 1 and 2 distinguish between the parameters of the two circuits then

$$y_{11} = y^1_{11} - \frac{y^1_{12} y^1_{21}}{y^1_{22} + y^2_{11}} \tag{9}$$

$$y_{12} = -\frac{y^1_{12} y^2_{12}}{y^1_{22} + y^2_{11}} \tag{10}$$

$$\mathrm{Y21} = -\frac{y^1_{21} y^2_{21}}{y^1_{22} + y^2_{11}} \tag{11}$$

$$y_{22} = y^2_{22} - \frac{y^2_{21} y^2_{12}}{y^1_{22} + y^2_{11}} \tag{12}$$

$$i_{s1} = i^1_{s1} - \frac{y^1_{12}}{y^1_{22} + y^2_{11}}(i^1_{s2} + i^2_{s1}) \tag{13}$$

$$i_{s2} = i^2_{s2} - \frac{y^2_{21}}{y^1_{22} + y^2_{11}}(i^1_{s2} + i^2_{s1}) \tag{14}$$

In fact, series combination for our three terminal networks is slightly more general than for resistors because the the junction of the two series networks may also include a third network to ground. Thus circuits such as the one in Figure 8 [8] (which is not a leaky tree) can be solved via this generalized **series–parallel** reduction.

[8]For the sake of clarity, we omit the third, grounded, terminal

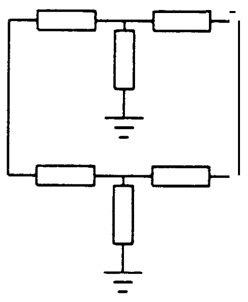
Figure 8: General series/parallel combination.

6 AC Analysis of Clusters

6.1 Regions of Linearity

The addition of piecewise linear devices complicates waveform evaluation because we must detect when devices cross regions of linearity. Thus the algorithm for waveform estimation becomes:

1. Using the current network state, compute device admittance parameters.
2. Compute waveform estimates for the linearized network.
3. Using the waveforms determine if and when each device changes regions of linearity.
4. If no device changes regions of linearity we are done. Otherwise advance to the time of the first region change and goto step 1.

Note that each of the inequalities in Equation 4 is of the form $f(x) > 0$ where f is a linear function of the terminal voltages. Since the terminal voltages vary with time, each function defines a time varying waveform, the "boundary" waveform. Since each of these functions must remain greater than zero if we are to remain in the current region of linearity, a change of the region of linearity is found by starting at the current time and searching forward for the next root of each boundary waveform.

There are at least two different approaches for computing boundary waveforms. One is to form the linear superposition of the terminal waveforms. However, the resulting waveform will contain all the poles from all of the terminal waveforms. Instead, in order to reduce the order of boundary waveforms we first compute their ***moments*** by linear superposition of the moments of the terminal voltages. Separate low order waveform approximations are then made from these moments.

6.2 Inter-cluster Waveforms

Another consequence of piecewise linear transistor models is that waveforms become piecewise exponential. For example

$$v(t) = \begin{cases} a_1 \epsilon^{-t/\tau_1} & 0 \leq t \leq t_1 \\ a_2 \epsilon^{-(t-t_1)/\tau_2} & t_1 \leq t \leq \infty \end{cases} \tag{15}$$

This is undesirable for cluster outputs because we would like to choose time steps for different clusters independently in order to efficiently simulate *stiff* circuits, that is circuits in which the time constants of different clusters vary widely.

However, moments are essentially the coefficients of the series expansion in s of the Laplace transform. The Laplace transform for the entire transient can be computed by summing the Laplace transforms of each of the pieces. Then, a continuous multipole approximation can be computed from the sum. To illustrate, note that $v(t)$ from our example can be expressed as the sum of time shifted exponentials

$$v(t) = a_1 \epsilon^{-t/\tau_1} u(t) - a_1 \epsilon^{t_1/\tau_1} \epsilon^{-(t-t_1)/\tau_1} u(t - t_1) + a_2 \epsilon^{-(t-t_1)/\tau_2} u(t - t_1) \tag{16}$$

where $u(t)$ is the unit step and the τ's may be complex. But if $F(s)$ is the Laplace transform of a time function $f(t)u(t)$, then the product $\epsilon^{-st_1}F(s)$ is the transform of the delayed function $f(t - t_1)u(t - t_1)$. Thus the series expansion in s of the Laplace transform of each time shifted exponential can be computed by multiplying the expansion for the unshifted exponential

$$\mathcal{L}\left\{\epsilon^{-t/\tau_1}\right\} = \frac{\tau_1}{s\tau_1 + 1} = \tau_1 \left(1 - s\tau_1 + (s\tau_1)^2 - (s\tau_1)^3 + \cdots\right) \tag{17}$$

by the expansion of ϵ^{-st_1}

$$\epsilon^{-st_1} = 1 - st_1 + \frac{(st_1)^2}{2!} - \frac{(st_1)^3}{3!} + \cdots \tag{18}$$

Because we only need lower order moments we need compute only the low order coefficients. The waveform approximation is computed from the low order moments.

6.3 Floating Capacitors

Switch level simulators are not able to deal with floating capacitors. Here, a floating capacitor is ***treated as*** a ***bidirectional*** coupling which requires the simultaneous evaluation of both terminals.

As outlined in Section 3 the computation of the moments proceeds by replacing capacitors with current sources. However, instead of inserting a single "floating" current source we insert two "grounded" current sources (Figure 9). When computing the k + lth moments, the current of an inserted current source becomes

$$i_c = C(M_{p_k} - M_{m_k}) \tag{19}$$

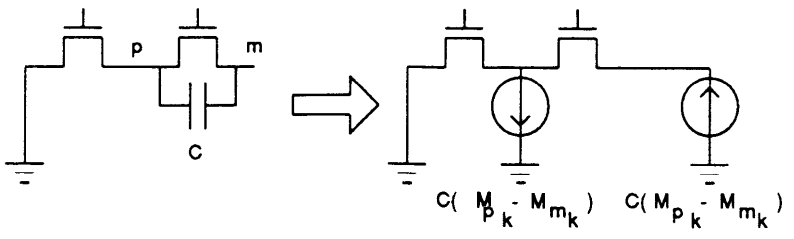

Figure 9: Floating capacitor connecting same cluster.

where M_{p_k} and M_{p_k} represent the kth moments of voltages of the nodes connected to the plus and minus capacitor terminals, respectively.

If both terminals are connected to the same cluster then the moment computation proceeds as for grounded capacitors. However, if the capacitor links two otherwise disconnected clusters (Figure 10) then the moments for both clusters must be com-

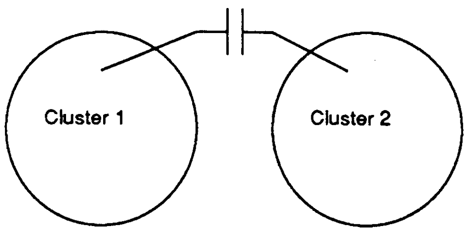

Figure 10: Floating capacitor connecting disconnected clusters.

puted in lockstep because the $k+1$th moments in each cluster depends upon the capacitor current which, in turn, is a function of the kth moments of nodes ***in both*** clusters. Thus the -1th moments are computed for Cluster 1 and Cluster 2, followed by the 0th moments for Cluster 1 and Cluster 2, etc. Lastly, if the capacitor connects two clusters which are also connected via inputs and outputs (Figure 11) then not

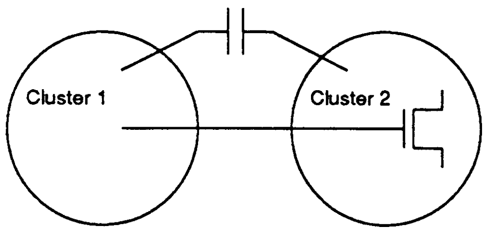

Figure 11: Floating capacitor connecting connected clusters.

only must moments be computed in lock step, but we must compute the kth moments of Cluster 1 ***before*** computing the kth moments of Cluster 2 because the unidirectional DC coupling causes the latter to be a function of the former. Note that in this case, from the standpoint of event scheduling the gate terminal in Cluster 2 is no longer treated as a cluster input.

In general we refer to a set of clusters coupled by floating capacitors as a group. Inputs are considered to be directed edges between clusters and the group is represented by a directed graph[9]. If the directed graph is cycle free then the correct order for the evaluation of moments ***can be*** found via a ***topological sort*** of the graph [10]. If the directed graph has cycles then the group has feedback and no topological sort exists. Feedback can be handled using ***tearing as described in*** a following section.

7 Relaxing Network Restrictions

In the preceding discussion we placed two restrictions on the network in order to guarantee that it can be solved with complexity $O(n)$: the topology must be a leaky tree and there must be no feedback However, many ICs contain circuits whose operation depends crucially upon one of these restrictions. If those circuits are small and if there aren't many of them, it may be practical to handle just those circuits using a more general, albeit less efficient, extension of the original algorithm. In this section we will show that the circuit decomposition technique known ***as tearing*** [11] can be used to to remove these two restrictions.

7.1 Non-Leaky Tree Topologies

Pillage and Dutta [PD90] ***used branch tearing*** to find the DC solution of resistor networks which ***are nearly trees. They*** point out that if the cotree [12] size is bounded, then the network may be solved in what is effectively linear time. We generalize that approach to solve nearly leaky trees of our more general three terminal devices.

Suppose we are given a circuit which is not a leaky tree. That circuit can be made into a leaky tree by finding a spanning tree and then disconnecting one terminal of ***each*** of ***the devices in the*** cotree. That ***is, we*** simplify the ***original*** network by ***tearing*** out the piece of wire which connects one terminal of each cotree device. However, rather than simply removing that wire, we replace it with an independent current source (Figure 12). Note that this augmented circuit can be solved using leaky tree analysis.

The solution of the original circuit can be found using multiple solutions of the augmented tree. First the inserted current sources are set to zero and the network is solved to find the voltages across each of them. Denote the resulting voltage across

[9]Note that while a cluster's graph is, by definition, connected, a group's directed graph may not he.

[10]A topological sort can be performed by depth *first search* with complexity $O(n)$ [AHU85].

"Circuit tearing was originally introduced by Kron[Kro39]. A more recent paper by Rohrer[Roh88] presents a delightfully simpk and intuitive interpretation of *tearing*.

[12]The cotree of a graph is defined as the set of all edges which are not members of the given spanning m e .

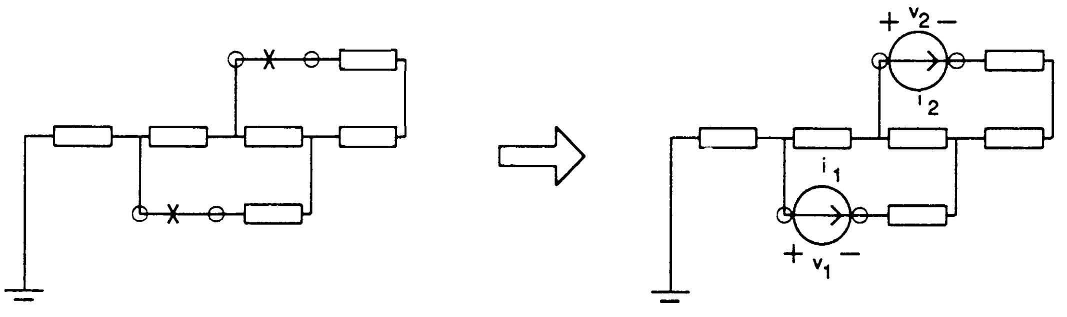

Figure 12: Tearing of non–leaky tree.

the kth inserted source by v_{o_k}, that is the response due to sources that were part of ***the original*** network. Next, set all sources (original and inserted) to zero. Then for each of the inserted sources, set only that source (let's say it is the kth source) to some nonzero constant i_{a_k} and solve the network for the voltages across each of the inserted current sources. Denote the ratio of the voltage across the jth inserted source to i_{a_k} by the transfer resistance $r_{j\,k}$. Then by superposition, the total response due to the original sources and with arbitrary settings for the inserted sources is given by (assuming n inserted sources)

$$\begin{bmatrix} v_{t_1} \\ v_{t_2} \\ \vdots \\ v_{t_n} \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{n1} & r_{n2} & \cdots & r_{nn} \end{bmatrix} \begin{bmatrix} i_{a_1} \\ i_{a_2} \\ \vdots \\ i_{a_n} \end{bmatrix} + \begin{bmatrix} v_{o_1} \\ v_{o_2} \\ \vdots \\ v_{o_n} \end{bmatrix} \tag{20}$$

If we set $v_t = 0$ in the above equation and solve for i_a we get the actual currents flowing through the tom wires of the original circuit (ie before augmentation). Finally, if we solve the augmented circuit with those currents we get the DC solution of the rest of the original circuit. Note that although this procedure requires the LU factoring of the n x $\boldsymbol{n}$ transfer resistance matrix, in general we expect $\boldsymbol{n}$ to be much smaller than the total number of branches in the network.

7.2 Feedback

Feedback is handled using procedures analogous to those applied to non-leaky tree topologies. Instead of tearing out edges to eliminate loops in a cluster's (undirected) graph, we tear out edges in order to eliminate cycles from a group's directed graph (Figure 13 (a)). The only additional complication arises when it is not possible to replace the tom wire with a current source because such a current source would see an infinite impedance. This commonly occurs when cluster inputs are only connected

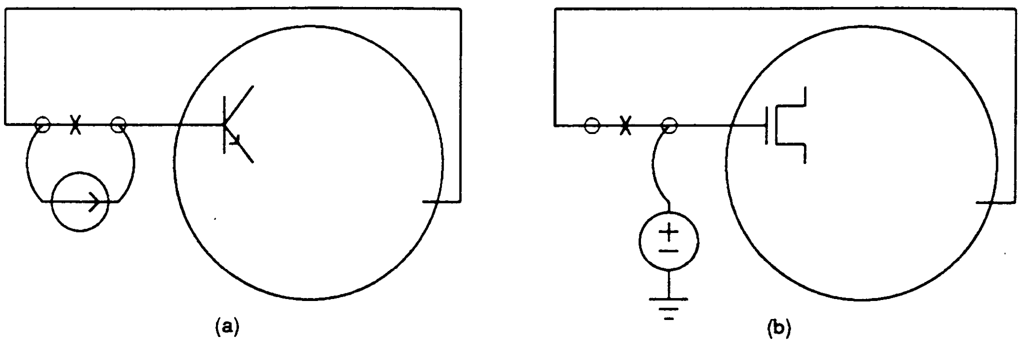

Figure 13 : Tearing of feedback.

to MOS gates. In this case we tear out the wire and drive the infinite impedance side with a grounded DC voltage source. The analysis then proceeds as described above except that operations on inserted current sources are performed on inserted voltage sources as well.

8 Conclusions

We have described a timing analysis algorithm which can achieve the efficiency of ***RC tree analysis*** while retaining much of the generality of ***AWE.*** The key problem that needed to be solved was the efficient computation of the moments of a circuit. While the Tree-Link analysis of AWEsim is general, it is inefficient ($O(n^2)$) for RC trees. On the other hand while RC tree analysis is efficient ($O(n)$) it only applies to RC trees.

We present an algorithm derived from RC tree analysis which can handle piece-wise linear transistor models connected in leaky tree topologies. For simple switch level models the complexity remains $O(n)$ for a given circuit state. Additionally, we show that the algorithm generalizes to include non-leaky tree topologies and feed-back, although the complexity is no longer $O(n)$ for those circuits. We intend to incorporate this timing analysis into a variable accuracy switch level simulator. We expect its speed to be comparable to switch level simulation for those portions of the circuit which utilize simple switch level models. However, the simulator should be able to simultaneously simulate more complex models and circuits with reduced efficiency for just the complex circuits.

References

[AHU85] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. ***Data Structures*** *and Algorithms.* Addison-Wesley, 1985.

[BS65] Amar *G.* Bose and Kenneth *N.* Stevens. ***Introductory Network Theory.*** Harper & Row, New York, 1965.

[Chu88] Chomg-Yeong Chu. ***Improved Models for Switch-Level Simulation.*** PhD thesis, Stanford University, November 1988.

[CL75] Leon 0. Chua and Pen-Mm Lin. ***Computer Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques.*** Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1975.

[Elm48] W. Elmore. The transient response of damped linear networks with particular regard to wideband amplifiers. ***Journal of Applied Physics,*** 19:55–63, January 1948.

[Hor83] ***Mark Alan*** Horowitz ***Timing Models for MOS Circuits.*** PhD thesis, Stanford University, December 1983.

[HPR87] X. Huang, L. T. Pillage, and R. A. Rohrer. Talisman: A piecewise linear simulator based on ***tree/link*** repartitioning. In ***International Conference on Computer-Aided Design,*** pages 98–101, November 1987.

[HSV81] Gary D. Hachtel and Alberto L. Sangiovanni-Vincentelli. A survey of third-generation simulation techniques. ***Proceedings of the IEEE,*** 69(10):1264–1280, October 1981.

[KAHS88] Russell Kao, Bob Alverson, Mark Horowitz, and Don Stark. Bisim: A simulator for custom ecl ***circuits. In International Conference on Computer-Aided Design,*** pages 62-65, November 1988.

[Kro39] G. Kron. ***Tensor Analysis of Networks.*** Wiley, 1939.

[PD90] Lawrence T. Pillage and Santanu Dutta. A path tracing algorithm for asymptotic waveform evaluation of lumped rlc circuit delay models. In ***ACM workshop Tau 90***, August 1990.

[PR90] Lawrence T. Pillage and Ronald A. Rohrer. Aymptotic waveform evaluation for timing ***analysis. IEEE Transactions on Computer-Aided Design,*** 9(4):352–366, April 1990.

[Roh88] R. ***A.*** Rohrer. Circuit partitioning simplified. ***IEEE Transactions on Circuits and Systems,*** 35(1):2–5, January 1988.

[RPH83] Jorge Rubinstein, Paul Penfield, Jr., and Mark A. Horowitz. Signal delay ***in*** rc tree networks. ***IEEE Transactions on Computer-Aided Design,*** CAD-2(3):202–211, July 1983.

[Ter83] *C.* J. Terman. ***Simulation Tools for Digital LSI Design.*** PhD thesis, Massachusetts Institute of Technology, September 1983.

[vB87] W. M. G. van Bokhoven. Piecewise linear analysis and simulation. In A. E. Ruehli, editor, ***Circuit Analysis, Simulation and Design, 2,*** chapter 10. Elsevier Science Publishers B. V. (North-Holland), 1987.