TICER: Realizable Reduction of Extracted RC Circuits

Bernard N. Sheehan
Mentor Graphics, Wilsonville, OR

Abstract

TIme Constant Equilibration Reduction (TICER) is a novel RC reduction method tailored for extract/reduce CAD tools. Geometry-minded extraction tools fracture nets into parasitics based on local changes in geometry. The resulting RC circuits can have a huge dynamic range of time-constants; by eliminating the extreme time-constants, TICER produces smaller, less-stiff RC networks. It produces realizable RC circuits; can retain original network topology; scales well to large networks (~10^7 nodes); preserves dc and ac behavior; handles resistor loops and floating capacitors; has controllable accuracy; operates in linear time on most nets.

Introduction

Rigorous accounting of metal R and C parasitics is desirable during final IC timing verification, especially in deep-submicron designs, because of the large delays attributable to interconnect. Fortunately, there has been striking progress in RC reduction technology during the last decade, beginning with AWE by Pileggi and Rohrer in 1990 [1]. The dominant trend has been towards projection techniques; see [2][3][4][5][6][7].

However, a disadvantage of most published methods is that the reduced circuit is *not* realizable. True, the methods preserve *electrical* properties at the ports. But what starts as an RC network becomes a mathematical macro-model--a set of poles and residues, for example. A related drawback is that the resulting macro-models presume special, often non-standard simulator capabilities [8][9].

Meanwhile, the problem of *realizable* reduction—that is, RC-in-RC-out methods--has been largely neglected in the literature. [10] describes how to realize a driving point admittance as an RC Cauer filter; but the ideas have not been extended to multiports. [11] presents a reduction scheme in which subcircuits are progressively merged (using an algorithm reminiscent of [12]) so as to match zero and first order moments. A weakness here is the lack of a mechanism for error control.

Unsatisfied with these limitations, we set out to devise a RC reduction technique that met the needs of commercial extraction/reduction. A practical method, we felt, had to:

1. yield a realizable RC network,
2. keep designated internal nodes,
3. scale to very-large (~10^7 element) networks,
4. preserve dc as well as ac characteristics,
5. control accuracy.

The outcome of our research is TICER--TIme Constant Equilibration Reduction. This novel approach uses 'time-constant equilibration' to achieve realizable reduction.

Nodal Time-Constants

Consider an N-terminal star network. The center of the star is node N, and the N terminals are labeled 0 to N-1 (0 being ground). A branch consisting of a conductance and capacitance in parallel--denoted by g_{iN} and c_{iN} for the i'th branch--joins each terminal to the central node. Some elements may be absent, in which case the corresponding g_{iN} or c_{iN} is zero.

The response of the central node when a step voltage is applied to the i'th terminal, all other terminals being grounded, is given by

$$h_{iN}(t) = \frac{g_{iN}}{g_N} + \left(\frac{c_{iN}}{c_N} - \frac{g_{iN}}{g_N} \right) e^{-t/t_N} \qquad (1.1)$$

where

$$g_N = \sum_{k=0}^{N-1} g_{kN}, \quad c_N = \sum_{k=0}^{N-1} c_{kN}, \quad t_N = \frac{c_N}{g_N} \qquad (1.2)$$

The general response of node N to arbitrary signals at the N-1 other terminals can be found by superposition of equations like (1.1). For our purposes, the key observation is that node N responds with a characteristic time constant

$$t_N = \sum_{k=0}^{N-1} c_{kN} \Big/ \sum_{k=0}^{N-1} g_{kN} \qquad (1.3)$$

and this time-constant is independent of which neighbor or combination of neighbors is agitated. We may therefore speak of the **time constant of a node** in a circuit. The time-constant of a node is the total capacitance from the node to other nodes and to ground divided by the sum of conductances from the node to other nodes and ground.

Consider now an arbitrary RC network which we want to reduce so as to preserve its behavior within a certain frequency range. We classify each node of a circuit into one of three categories according to whether a node's time-constant is less than, greater than, or between the min and max time-constants defining the frequency range of interest.[1] We refer to a node as **quick**, **slow**, or **normal** according to this criteria.

The importance of this classification comes from the fact that both quick and slow nodes can be eliminated from the network without significantly altering its behavior--at least not in the frequency range of interest. We call this process--reducing a circuit to a smaller one by eliminating quick and slow nodes--**time constant equilibration**. Whereas the time-constants in the original circuit may span a very wide dynamic

[1] The conversion of a range of time-constants to a range of frequencies, of course, can only be done approximately, using an expression like $f=4/\tau$. Great precision is not required here.

0-7803-5832-X /99/$10.00 ©1999 IEEE.

range, those in the equilibrated circuit will be clustered around the time-scale (frequency range) of interest.

Node Elimination

In this section we present the detailed procedure for eliminating quick and slow nodes. We arrive at our goal most easily by beginning from the nodal equations of an RC circuit in the Laplace domain:

$$(Cs+G)v = Yv = J \qquad (2.1)$$

where s is the complex frequency, $C \in R^{NxN}$ and $G \in R^{NxN}$ are the nodal capacitance and conductance matrices, $v \in R^N$ is the vector of nodal voltages, and $J \in R^N$ is the vector of current sources at the nodes.

For simplicity, assume that the node we wish to eliminate is the last node N. Writing (2.1) as a block system

$$\begin{bmatrix} \tilde{Y} & y \\ y^T & (g_N + sc_N) \end{bmatrix} \begin{bmatrix} \tilde{v} \\ v_N \end{bmatrix} = \begin{bmatrix} \tilde{J} \\ j_N \end{bmatrix} \qquad (2.2)$$

we can solve for v_N from the second block equation and substitute it into the first block equation to obtain

$$(\tilde{Y} - E)v_N = \tilde{J} - F \qquad (2.3)$$

$$E_{ij} = \frac{(g_{iN} + sc_{iN})(g_{jN} + sc_{jN})}{g_N + sc_N} \qquad (2.4)$$

$$F_i = \frac{(g_{iN} + sc_{iN})}{g_N + sc_N} j_N \qquad (2.5)$$

In these equations γ_N and χ_N are defined analogously to our previous discussion, i.e.,

$$g_N = \sum_{k=0}^{N-1} g_{kN}, \quad c_N = \sum_{k=0}^{N-1} c_{kN} \qquad (2.6)$$

with some terms in the sums possibly being zero. We also assume there is no current source at the eliminated node, so that j_N and F are zero.

Quick Nodes

First suppose $sc_N << g_N$; the eliminated node is a quick node, and we approximate the element E_{ij} arising from elimination by

$$E_{ij} \approx \frac{g_{iN} g_{jN}}{g_N}\left(1 - \frac{sc_N}{g_N}\right) + s\frac{g_{iN} c_{jN} + g_{jN} c_{iN}}{g_N} \qquad (3.1)$$

this expression being obtained from (2.4) by expanding in powers of s and retaining terms up to the first order. The second term in the parentheses is negligible by assumption; hence,

$$E_{ij} \approx \frac{g_{iN} g_{jN}}{g_N} + s\frac{g_{iN} c_{jN} + g_{jN} c_{iN}}{g_N} \qquad (3.2)$$

This last equation can be translated into a procedure for physically modifying the circuit. To eliminate a quick node N from a network, first remove all resistors and capacitors connecting other nodes to node N. Then insert new resistors and capacitors between former neighbors of N according to the following two rules. If nodes i and j had been connected to N through conductances g_{iN} and g_{jN}, insert a conductance $g_{iN} g_{jN} / g_N$ from i to j; if node i had a capacitor c_{iN} to N, and node j had a conductance g_{jN} to N, then insert capacitor $c_{iN} g_{jN} / g_N$ between i and j. It is easy to verify that the admittance matrix of the circuit thus modified is given by (2.3) and (3.2). Remember in applying these rules that ground is treated just like other neighbors of N.

The above elimination rules corresponds to the condition that node N in the original circuit is always **fully relaxed**--that is, its voltage at all times is determined by being in dc equilibrium with its neighbors, i.e.

$$v_N(t) = \frac{1}{g_N}\sum_{j=0}^{N-1} g_{jN} v_j(t) \qquad (3.3)$$

If this condition holds, the current flowing from node i to node N through g_{iN} can be shown to equal

$$i_{iN}(t) = \sum_{j=0}^{N-1} \frac{g_{iN} g_{jN}}{g_N}[v_i(t) - v_j(t)] \qquad (3.4)$$

Similarly, when the central node is fully relaxed, the net charge on a capacitor c_{iN} can be shown to equal

$$q_{iN}(t) = \sum_{j=0}^{N-1} \frac{c_{iN} g_{jN}}{g_N}[v_i(t) - v_j(t)] \qquad (3.5)$$

Interpreting each term in (3.4) and (3.5) as a resistor or capacitor between i and j leads again to the elimination rules described above.

Slow Nodes

Next suppose $sc_N >> g_N$; the eliminated node is a slow node, and we approximate the element E_{ij} arising from elimination by

$$E_{ij} \approx \frac{g_{iN} g_{jN}}{g_N} + s\frac{c_{iN} c_{jN}}{c_N} \qquad (4.1)$$

This equation comes from substituting the expansion

$$\frac{1}{g_N + sc_N} \approx \frac{1}{sc_N}\left(1 - \frac{g_N}{sc_N}\right) \qquad (4.2)$$

into (2.4) and retaining terms containing s; however, to preserve dc characteristics, $g_{iN} g_{jN} / g_N$ is used in place of whatever constant terms come from the expansion.

From this we get the following slow-node elimination procedure. To eliminate a slow node N from a network, first remove all resistors and capacitors connecting any nodes to node N. Then, as before, if nodes i and j had been connected to N through conductances g_{iN} and g_{jN}, insert conductance $g_{iN} g_{jN} / g_N$ from i to j; if node i had a capacitor c_{iN} to N, and node j had a capacitance c_{jN} to N, insert capacitor $c_{iN} c_{jN} / g_N$ between i and j.

These slow-node rules corresponds to the condition that node N in the original circuit is, from the point of view of the capacitors, a **floating node**.

Properties of TICER

The method of reducing an RC circuit by successively finding quick or slow nodes in the circuit and then eliminating

them by the preceding rules is called **TIme Constant Equilibration Reduction** or **TICER**. Mostly quick nodes are eliminated in practice. Nodes that are neither quick nor slow are left in tact, as are nodes with too many neighbors or those designated to be kept by the users--so called **fixed** nodes.

TICER has a number of desirable properties. Foremost, perhaps, is the property that the reduced circuit is a *realizable* RC circuit; the desirability of this in a parasitic extraction setting is clear since it cannot be assumed that the down-range simulator can handle special macro-models. Being an RC circuit, the output is passive, so stability is also not an issue. Resistor loops or floating capacitors are not problematic. DC characteristics are exactly preserved; this is important for calculating operating points and load currents when connecting to non-linear devices. Total capacitance to ground or between nets is also preserved, under lax assumptions.[2] Thus, down-range tools that use total capacitance for driver-table loading calculations or for crosstalk estimates get the same answer from the reduced and original circuits. Finally, judicious assignment of fixed nodes can preserve essential network topology.

These features collectively argue that TICER might well be the tool of choice for general-purpose parasitic extraction applications.

Because the closest counterpart to TICER in the literature is probably the S-parameter method in [11], it may help to compare these more closely. Both methods produce realizable circuits. Both proceed by eliminating internal nodes of subcircuits. Both match, or nearly match, zero and first order moments. (One matches S-parameter moments, the other Y-parameter moments; but this is equivalent). Such are the similarities.

But the differences--especially in the way the two methods view the circuit--are far reaching. [11] views the circuit as a dynamic set of multiport sub-networks, pairs of which are progressively merged. TICER, by contrast, focuses on a particular node and its neighbors, temporarily regards these as an multiport, eliminates the internal node, then returns to a flattened view of the circuit before selecting another node for elimination. This leads to a simpler data structure--adjacency lists versus a list of multiports. More importantly, TICER's approach controls accuracy through its choice of the time-constant boundaries that separate quick, normal, and slow nodes. Wider boundaries mean greater accuracy but less reduction; narrower boundaries mean the opposite. [11] blindly follows geometry.

Preservation of Elmore Delays

TICER has another elegant property: it preserves Elmore delays bidirectionally through RC ladders--at least if one eliminates only internal, quick nodes. To see this, consider a R_1-C-R_2 T network. If you remove the central node using the quick-node rules presented earlier, the result is a π network with capacitive legs $R_2C/(R_1+R_2)$ and $R_1C/(R_1+R_2)$ and a bridge resistance R_1+R_2. For this circuit, the Elmore delay from node 1 to 2 is R_1C and the delay from 2 to 1 is R_2C; but these are the same as the Elmore delays of the original T network. By induction, the Elmore delays of arbitrary RC ladders are unchanged when any or all internal quick nodes are eliminated.

Examples

Having laid out the method and its more important properties, let us now look at a few examples in order to be able to judge its effectiveness in practice.

Figure 1 plots the distribution of nodal time-constants before and after reduction for a small IC design composed of 397 nets/5650 nodes. Coupling capacitance was not included in the extraction. The original design had nodal time constants spanning more than 8 decades!

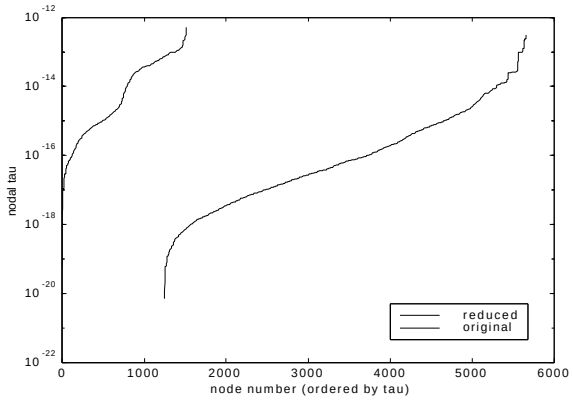

Figure 1 Distribution of nodal time-constants

Looking at the figure, we see how TICER's equilibration process has removed most of the extremely small time constants. For instance, before reduction, the design had 3721 nodes with time constants at or below 10^{-16} seconds; after reduction, only 121 such nodes. (About 1100 nodes in the original circuit were purely 'resistive' nodes with τ=0.0; these, all of which were eliminated by TICER, do not appear in the plot.)

Table 1

net size (nodes)	number of nets	percent reduction	number of ports	avg. peak error (V)
1-5	75	30%	2.44	0.00011
6-10	121	60%	3.16	0.00131
11-20	138	72%	3.92	0.00256
21-40	42	81%	4.59	0.00489
41-100	21	86%	5.76	0.00645

For this same design, Table I gives average statistics for nets in various size ranges. For example, there were 42 nets in the design whose sizes were between 21 and 40 nodes. Of these nets, the average reduction in node count was 81%; the nets had an average of 4.59 ports (ports were fixed and could not be eliminated); the average peak error when original and reduced nets were simulated side by side was 4.89 mV, when a 1V, 0.2ps ramp was applied to one of the ports.

[2] The stipulation for capacitance to ground or between nets to be preserved is that (1) slow nodes and (2) nodes with shunt resistances to ground are not eliminated.

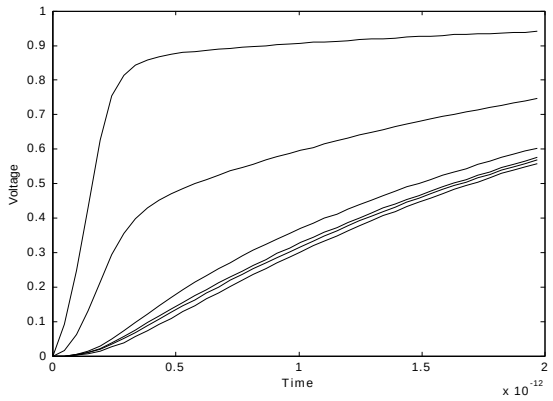

Figure 2. Response of a 96-node network

Figure 2 plots the response of the largest net in the design, superimposing the waveforms of the net before and after reduction. The before and after waveforms are optically indistinguishable--differing by a maximum of 0.0012V. TICER reduced the net from 96 to 17 nodes.

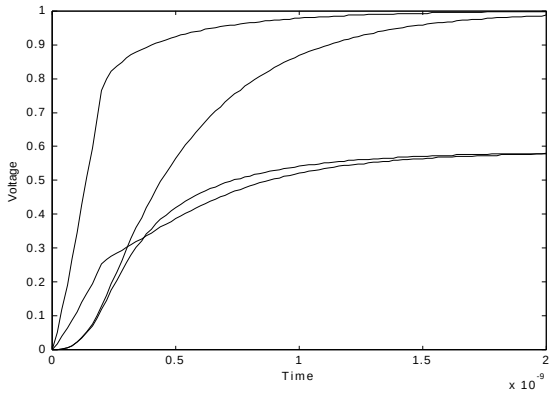

Figure 3. Coupled RC lines

Although our sample design did not include coupling capacitance, TICER also works well with coupled nets. To see this, consider a 20-section capacitively coupled RC ladder network representing two neighboring 1cm lengths of metal 4 traces in 0.18 micron technology [13]. Figure 3 compares the simulated waveforms of the 42 node original (solid) and 12 node reduced(dashed) networks. Again, the results are optically almost indistinguishable.

Conclusion

Since TICER sequentially eliminates one node at a time, optionally giving preference to nodes of low degree, the calculations involve only local modifications of the network graph. Hence, the run-time performance is O(n) on sparse graphs, where n is the number of nodes eliminated.

TICER is particularly suited for use with geometric extraction tools. These tools are triggered to insert nodes wherever geometry changes--at corners, vias, or spacing shifts. The granularity of extraction, therefore, can be quite irregular, with many minute parasitics and relatively few larger ones. Guided by nodal time-constants, TICER systematically pinpoints the smallest parasitics and eliminates them, thereby compacting the netlist into a more uniformly fractured, less stiff, circuit. This is done with little loss in accuracy.

In our view, TICER complements rather than replaces other reduction methods (e.g. those based on orthogonal projection using moments). Other methods may produce higher compression in some cases. TICER, for example, is not particularly effective on dense meshes, since as mesh junctions are eliminated, the degrees of remaining nodes increases such that little overall compression occurs. On most other topologies, however, TICER works excellently. Its salient advantage is this: RC-in, RC-out. Gone is the burden of non-portable macro-models. If realizable circuits are needed, or controllable accuracy, or approximate topology preservation or preservation of total coupling or grounding capacitance, then TICER is the clear choice.

[1] L. Rohrer and L. Pillage, "Asymptotic Waveform Evaluation for Timing Analysis," *IEEE Trans. Computer Aided Design*, vol. 9, pp. 352-66, 1990.

[2] P. Feldmann and R. W. Freund, "Reduced-Order Modeling of Large Linear Subcircuits via a Block Lanczos Algorithm," *32nd DAC*, pp. 474-79, 1995.

[3] K. Kerns, I. Wemple, A. Yang, "Stable and Efficient Reduction of Substrate Model Networks Using Congruence Transforms," ICCAD 1995, pp. 207-14.

[4] M. Silveira, M. Kamon, I. Elfadel and J. White, "A Coordinate-Transformed Arnoldi Algorithm for Generating Guaranteed Stable Reduced-Order Models of RLC Circuits," *33rd DAC*, pp. 288-94, 1996.

[5] B. Sheehan, "ENOR: Model Order Reduction of RLC Circuits Using Nodal Equations for Efficient Factorization," 36th DAC, pp17-21, 1999.

[6] A. Odabasioglu, M. Celik, L. T. Pileggi, "PRIMA: Passive Reduced-order Interconnect Macromodeling Algorithm," *34th DAC*, pp. 58-65, 1997.

[7] B. Sheehan, "Projective Convolution: RLC Model-Order Reduction Using the Impulse Response", DATE-99, 1999.

[8] V. Raghavan, J. E. Bracken, R. A. Rohrer, "AWESpice: A general tool for the accurate and efficient simulation of interconnect problems," *29th DAC*, pp. 87-92, 1992.

[9] Y. Kim, N. Gopal, and L. T. Pillage, "Time-Domain Macromodels for VLSI Interconnect Analysis", *IEEE Trans. on CAD*, vol. 14, No. 10, pp. 1257-70, Oct. 1994.

[10] R. W. Freund and P. Feldmann, "Reduced-Order Modeling of Large Passive Linear Circuits by Means of the SyPVL Algorithm", *33rd DAC*, pp. 280-87, 1996.

[11] H. Liao and W. W. Dia, "Partitioning and Reduction of RC Interconnect Networks Based on Scattering Parameter Macromodels", ICCAD 1995, pp. 704-09.

[12] V. A. Monaco and P. Tiberio, "Automatic Scattering Matrix Computation of Microwave Circuits", *Alta Freq.* Vol 39, Feb 1970, pp. 59-64.

[13] J. Cong, Z. He, C. Koh, K. Khoo, "Interconnect Design for Deep Submicron ICs", ICCAD-97, pp. 478-85.