

## Realizable Reduction of Interconnect Circuits Including Self and Mutual Inductances

Chirayu S. Amin, Masud H. Chowdhury, and Yehea I. Ismail

**Abstract**—Reduction of an extracted netlist is an important preprocessing step for techniques such as model order reduction (MOR) in the design and analysis of very large scale integration circuits. This paper describes a method for realizable reduction of extracted resistance–capacitance–inductance–mutual inductance netlists by node elimination. The method is much faster than MOR techniques and, hence, is appropriate as a preprocessing step. The proposed method eliminates nodes with time constants below a user-specified time constant. By giving the freedom to the user to select a critical point in the spectrum of nodal time constants, this method provides an option to make a tradeoff between accuracy and reduction. The proposed method preserves the dc characteristics and the first two moments at all nodes. It also recognizes and eliminates all the redundant inductances generated by the extraction tools. The proposed method naturally reduces to TICER (Sheehan, 1999) in the absence of any inductances.

**Index Terms**—Circuit, circuit reduction, estimation, interconnect, model order reduction (MOR), simulation, timing verification, very large scale integration (VLSI).

### I. INTRODUCTION

With increasing frequencies and faster signal transition times, on-chip inductive effects are becoming increasingly important [1], [2]. Consequently, many commercial and proprietary extraction tools, such as [3], generate resistance–capacitance–inductance–mutual inductance (RLCK) circuits for high-performance designs. Due to the large amount of data typically generated by extraction tools, significant runtime and memory issues affect the analysis tools. Therefore, in the past decade, there has been a significant focus on model order reduction (MOR) methods, which attempt to model the extracted netlist by a smaller model with minimal loss in accuracy. Starting with asymptotic waveform evaluation (AWE) [5], methods such as [6] and [7] use moment matching—either explicitly or implicitly—and projection techniques to generate a low-order approximation of the original circuit. More recently, PRIMA [8] modified these techniques to guarantee passivity of the reduced circuits.

Reducing an extracted netlist is an important step before the netlist is fed into tools such as AWE and PRIMA so that they run faster. In order to take advantage of the MOR techniques, it is important to have a realizable netlist reduction method, i.e., the RLCK in–RLCK out feature. Realizability is important because this avoids the modification of mainstream analysis tools to handle reduced state-space or transfer function representations [9], since these tools are usually geared toward reading circuit netlists. Moreover, some analysis tools, such as circuit checkers, can only accept inputs in terms of RLCK circuits. In addition to realizability, however, maintaining sparsity is also important when reducing a circuit with a large number of ports. Even though the original circuit is large, it is very sparse since each node is only connected to a few

nodes. It is important to maintain sparsity since MOR techniques perform better on sparse circuits.

Circuit reduction approaches based on Gaussian elimination include [10] and [11]. A different approach based on Gaussian elimination called TICER was presented in [12] for RC circuits. By selectively removing the nonport nodes of the original circuit, a smaller, realizable, passive RC circuit is produced. A method to reduce resistance–inductance–capacitance (RLC) netlists with coupling–inductors was presented in [13], but it requires matrix inversion and long runtime. In this paper, we extend the TICER approach to reduce general RLC circuits including coupling–inductors using runtime in the order of the number of nodes in netlist. We also present a heuristic to control the sparsity of the reduced circuit. This is similar to the technique presented for RC circuits in [10].

Most of the MOR techniques require moments calculation, which is computationally very expensive for large RLCK netlists. Moments calculation requires inversion of large matrices which may or may not be sparse because of coupling between nodes. This is a process with superlinear complexity. Although the reduced models still have to be produced after a large netlist is converted to a smaller one using the proposed realizable reduction technique, MOR techniques will be much faster on the smaller, reduced circuit. Computing moments of the reduced circuit will be much faster because it requires inverting a much smaller system matrix, which has sparsity comparable to that of the original large RLCK netlist. Hence, the main contribution of the work presented here is at the preprocessing level, where MOR techniques and SPICE transient simulations are computationally very expensive in handling very large netlists. Once the netlist is preprocessed, there can be a significant reduction in run-time to produce reduced models from every source to every dependent sink or to simulate the circuit using SPICE.

The proposed method can also be easily embedded in extraction tools, most of which produce large parasitic netlists in the gigabytes size range, due to a lack of good realizable reduction techniques for RLCK netlists. These large parasitic netlists cause memory and runtime problems for many static timing-analysis tools. The method proposed in this paper solves the problem by reducing the size of netlists produced by extraction tools.

The presented reduction algorithm is much faster than MOR techniques and can be used as a preprocessing step for these techniques. The rest of the paper is organized as follows. The underlying theory behind the proposed reduction method is discussed in Section II. In Section III, a high-level algorithm for reducing RLC circuits is presented. Experimental results are provided in Section IV. Finally, Section V concludes the paper. The appendix describes the procedure to reduce RLC circuits with coupling–inductors.

### II. THEORY

Consider a source-free RLC circuit consisting of  $n$  nodes. The nodal voltages must satisfy the following equation in the  $s$ -domain:

$$Y(s)V(s) = 0 \quad (1)$$

where  $Y(s)$  is the  $n \times n$  admittance matrix. Consider the  $i$ th node of the circuit and its  $k$  neighbors as shown in Fig. 1(a). The  $i$ th row of (1) is given by

$$Y_i V_i - y_1 V_1 - y_2 V_2 - \dots - y_k V_k = 0, \quad \text{where } Y_i = \sum_{j=1}^k y_j. \quad (2)$$

Manuscript received May 5, 2003; revised December 18, 2003 and April 8, 2004. This work was supported in part by the National Science Foundation CAREER Award CCR-0237822, in part by the Semiconductor Research Corporation under Contract 1073, and in part by gifts and grants from the Intel Corporation, IBM Corporation, and Multigig Inc. This paper was recommended by Associate Editor F. N. Najm.

The authors are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: c-amin@northwestern.edu; masud@ece.northwestern.edu; ismail@ece.northwestern.edu).

Digital Object Identifier 10.1109/TCAD.2004.840545

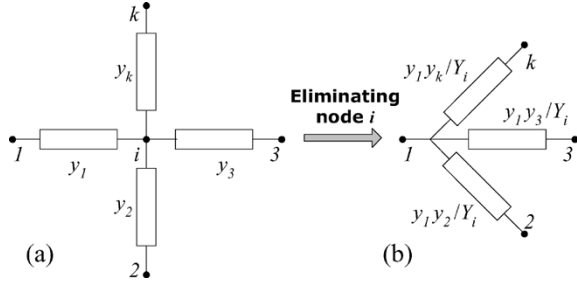


Fig. 1. General node in an  $RLC$  circuit. Admittances are added between node 1 and other neighbors of node  $i$  due to elimination of node  $i$ .

In order to eliminate  $V_i$  from the system (which is equivalent to eliminating the  $i$ th node), we solve for  $V_i$  using (2)

$$V_i = \frac{\left( \sum_{j=1}^k y_j V_j \right)}{Y_i} \quad (3)$$

and substitute for  $V_i$  in the  $k$  equations where  $V_i$  occurs. Consider the first neighbor of  $i$ . Its equation is now given by

$$\left( \hat{Y}_1 + y_1 - \frac{y_1^2}{Y_i} \right) V_1 - \frac{\left( \sum_{j=2}^k y_1 y_j V_j \right)}{Y_i} - \sum_{\substack{r=1 \\ r \neq i}}^{k-1} y_r V_r = 0$$

$$\hat{Y}_1 = \sum_{\substack{r=1 \\ r \neq i}}^{k-1} y_r \quad (4)$$

where  $\hat{Y}_1$  is the sum of all admittances from node 1 except to node  $i$  and  $k_1$  is the number of nodes connected to node 1. The above equation can be simplified to

$$\left( \hat{Y}_1 + \frac{\left( \sum_{j=2}^k y_1 y_j \right)}{Y_i} \right) V_1 - \frac{\left( \sum_{j=2}^k y_1 y_j V_j \right)}{Y_i} - \sum_{\substack{r=1 \\ r \neq i}}^{k-1} y_r V_r = 0. \quad (5)$$

Note that this is equivalent to adding  $k - 1$  new elements between node 1 and the  $k - 1$  former neighbors of node  $i$ , [see Fig. 1(b)]. Specifically, for any two neighbors of node  $i$ , say  $m$  and  $n$ , the elimination of node  $i$  results in the addition of a new element between nodes  $m$  and  $n$ , whose admittance is given by

$$y_{mn} = \frac{(y_m y_n)}{Y_i}. \quad (6)$$

Thus, repeating this process for all the  $k$  neighbors of  $i$  will result in the addition of  $k(k - 1)/2$  new elements. Note that the elimination of node  $i$  may introduce a *fill-in* in the original  $Y$  matrix of (1). For instance, referring to (6), if the  $(m, n)$ <sup>th</sup> entry in  $Y$  was a zero, *i.e.*, no element existed connecting  $m$  and  $n$ , elimination of node  $i$  would produce a fill-in in the  $(m, n)$ th entry of  $Y$ . In general, the formula for computing the fill-in,  $\phi_i$ , produced by eliminating node  $i$  with  $k$  neighbors is given by

$$\phi_i = k \frac{(k - 1)}{2} - k - p \quad (7)$$

where  $p$  is the number of elements connecting the neighbors of  $i$  in the original system.

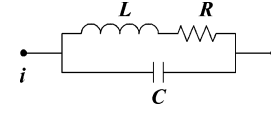


Fig. 2. Admittance branch connected to node  $i$ .

This type of nodal elimination preserves nodal voltages and equivalent branch admittance between nodes. Although the branch currents are not preserved explicitly, they are preserved implicitly because they only depend on nodal voltages and equivalent branch admittances, which are preserved. The new admittance given by (6) will be a polynomial in  $s$ . Without loss of generality for  $RLC$  circuits (circuits with  $K$  are discussed later), we assume that each admittance connecting a pair of nodes in the original system in Fig. 1 consists of a resistor ( $R$ ), inductor ( $L$ ), and a capacitor ( $C$ ) connected as shown in Fig. 2.

In most practical circuits, one or more of these elements will be zero. Note that this topology is general enough to handle any  $RLC$  circuit including coupling capacitances. If no inductance is incident on the node, then (6) reduces to the TICER case and we proceed as outlined in [12]. In that case (6) can be expressed by

$$y_{mn} = \frac{(g_m + sc_m)(g_n + sc_n)}{(G_i + sC_i)} \quad (8)$$

where  $G_i = \sum_{j=1}^k g_j$  is the sum of all conductances to node  $i$ , and  $C_i = \sum_{j=1}^k c_j$  is the sum of all capacitances to node  $i$ . For pure  $LC$  case (6) can be expressed as

$$y_{mn} = \frac{\left( \frac{b_m}{s} + sc_m \right) \left( \frac{b_n}{s} + sc_n \right)}{\left( \frac{B_i}{s} + sC_i \right)} \quad (9)$$

where  $B_i = \sum_{j=1}^k b_j$  is the sum of all susceptances (reciprocal inductances) connected to node  $i$ .

For every node in the circuit two time constants are defined: the  $RC$  time constant given by  $\tau_{RCi} = C_i/G_i$ , and the  $LC$  time constant given by  $\tau_{LCi} = \sqrt{C_i/B_i}$ . The nodal time constant at a node  $i$  is given by

$$\tau_i = \max(\tau_{RCi}, \tau_{LCi}). \quad (10)$$

A node  $i$  is said to be a *quick node* if

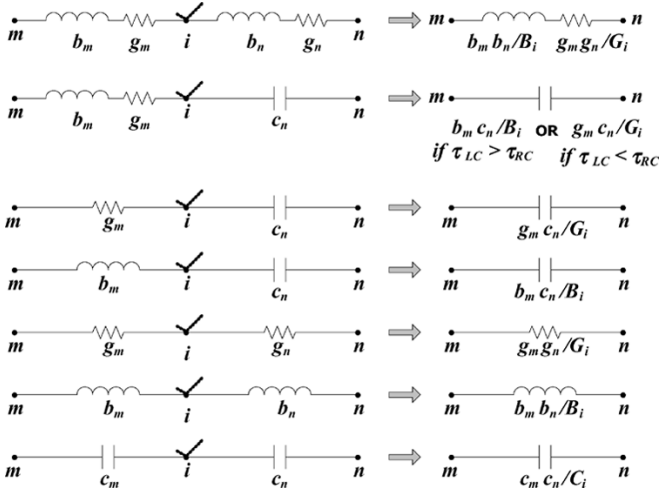
$$\tau_i < \tau_{\min} = \frac{2\pi}{\omega_{\max}} \quad (11)$$

where  $\tau_{\min}$  is a user defined time constant below which a node is considered quick and depends on the maximum frequency of interest in the circuit,  $\omega_{\max}$ , as given above. The selection of  $\tau_{\min}$  is circuit specific and is discussed in Sections III and IV. According to (10) and (11), both time constants of a quick node must be less than  $\tau_{\min}$ . Note that  $\tau_{\min}$  is proportional to  $1/s_{\max}$  as given by (11). Hence, a quick node satisfies the following approximations:  $sC_i < G_i$ ,  $G_i < B_i/s$ , and  $sC_i < B_i/s$ .

To eliminate a quick node with the above general  $RLC$  branch (Fig. 2) two cases can be considered. In the first case,  $\tau_{RCi}$  is much larger than  $\tau_{LCi}$  and (8) can be used as in TICER [12]. With the quick node approximation ( $sC_i < G_i$ ), (8) can be expanded into Taylor series up to first order

$$y_{mn} = \frac{g_m g_n}{G_i} + s \frac{c_m g_n + c_n g_m}{G_i} + \dots \quad (12)$$

Here, the constant term in (12) gives the required conductance to be inserted between node  $m$  and  $n$  to eliminate node  $i$ . The coefficient of  $s$  gives the capacitance value to be inserted. In the second case,  $\tau_{LCi}$

Fig. 3. Rules for eliminating node  $i$ .

is much larger than  $\tau_{RCi}$  and (9) is used. Again, with the quick node approximation ( $sC_i < B_i/s$ ), (9) can be expanded into Taylor series

$$y_{mn} = \frac{1}{s} \frac{b_m b_n}{B_i} + s \frac{c_m b_n + c_n b_m}{B_i} + \dots \quad (13)$$

The coefficient of  $1/s$  in (13) gives the required susceptance (reciprocal of inductance) to be inserted between nodes  $m$  and  $n$  to eliminate node  $i$ . The coefficient of  $s$  gives the capacitance value to be inserted.

For the general case, when both  $R$  and  $L$  are present with  $C$  as in Fig. 2, the rules for eliminating node  $i$  based on the (12) and (13) are shown in Fig. 3. The values for the conductance and the susceptance come directly from the (12) and (13), respectively. However, two expressions for capacitance are obtained from (12) and (13). If  $\tau_{RCi}$  is larger than  $\tau_{LCi}$ , the expression for capacitance is taken from (12), and when  $\tau_{LCi}$  is larger than  $\tau_{RCi}$ , the expression for capacitance is taken from (13). Therefore, all but one of the rules shown in Fig. 3 for merging any two branches connected to a particular node  $i$  can be easily derived from (12) and (13). The exception is for the case when nodes  $m$  and  $n$  are connected to node  $i$  through capacitances. In that case, the value of the capacitance to be inserted is  $c_m c_n / C_i$ , which is the series combination of the capacitances. Since only positive valued  $RLC$  elements are added during node elimination, the passivity of the reduced circuit is guaranteed by construction. In contrast to TICER, we do not consider the notion of *slow nodes*. This is because, in practice, the lowest frequency of interest is zero, i.e., the dc operation of the circuit is also required. In the following section, a high-level algorithm implementing these ideas is presented. The Appendix describes a method to reduce  $RLCK$  circuits, i.e.,  $RLC$  netlists with coupling-inductors.

It should be noticed that the derivation of the above method of node elimination is based on two approximations, namely dominant  $\tau_{RCi}$  and dominant  $\tau_{LCi}$  approximations. Therefore, the correctness of the method is dependent on the validity of these two approximations. If for any node the two time constants ( $\tau_{RCi}$  and  $\tau_{LCi}$ ) are far from each other, the proposed method will work very effectively. If the two time constants are comparable (which rarely happens in practice), then the reduction will introduce larger error. In such situation it is suggested not to eliminate that node if very low error is required. Therefore, to eliminate a node from the net list with less error two criteria have to be considered. First, the node has to be a quick node, and second the two time constants ( $\tau_{RCi}$  and  $\tau_{LCi}$ ) should not be comparable. Tuning the threshold for these two conditions produces a tradeoff between accuracy and reduction.

### REDUCE\_NETLIST ( $\tau_{\min}, \phi_{\max}$ )

1. Place all non-port nodes of the circuit in a priority queue sorted by local nodal time-constants.
2.  $i =$  head of the priority queue
3. while (queue is not empty AND  $\tau_i < \tau_{\min}$  AND  $\phi_i < \phi_{\max}$ )
4.   set  $S = \{\text{neighbors of } i\}$
5.   if ( $1/B_i = 0$ )
6.     eliminate  $i$  according to the TICER quick-node rules in [12]
7.   else
8.     eliminate  $i$  according to rules in Figure 3
9.   end if
10.   update time-constants of nodes in set  $S$  (neighbors of  $i$ )
11.    $i =$  head of the priority queue
12. end while

Fig. 4. Circuit-reduction algorithm.

### III. REDUCTION ALGORITHM

The high-level reduction algorithm is shown in Fig. 4. All the nodes in the original circuit are stored in a priority queue with the node with the smallest nodal time constant at the head of the queue. In addition to checking for the time constant of the node, we also check if the fill-in due to this node is less than some user specified threshold  $\phi_{\max}$ . If unspecified this threshold defaults to zero. This heuristic is added to ensure that the admittance matrix of the reduced circuit remains sparse. Of course, this heuristic may cause the algorithm to get stuck in a local optimum where even though a node's time constant may be less than  $\tau_{\min}$ , it may fail the fill-in check. However, for most interconnect circuits, such as those with a tree like topology, this algorithm will produce reasonably sparse reduced circuits. On the other hand, for very dense topologies, where each node is connected to many neighbors, the efficiency of reduction will be less because of the fill-in criteria and the algorithm will perform poorly. However, this same problem also occurs in TICER for  $RC$  circuits. The main contribution here is the extension of TICER to circuits with  $L$  and  $K$  elements and controlling the density of system matrix through the fill-in criteria. While not explicitly shown in Fig. 4, certain nonport nodes may also be marked as required by the user, in which case, these nodes cannot be eliminated. A check for this can be easily added to the algorithm.

One natural question arises regarding the choice of  $\tau_{\min}$ . Higher the value of  $\tau_{\min}$ , higher frequency components get eliminated and higher the loss of accuracy. While signal transition times and operating frequencies do play a part in the choice of  $\tau_{\min}$ , a histogram showing the distribution of the time constants can be very helpful. As we show in the next section, a large number of nodes typically have very small time constants. By choosing  $\tau_{\min}$  appropriately, these nodes can be eliminated with almost no loss in accuracy. A histogram helps in placing the time constants of all the nodes in the circuit in perspective as described in the next section. The main essence of this proposed method is speed. Making the method mathematically more robust and having a more robust technique to determine  $\tau_{\min}$  will only increase the complexity of calculation without significant improvement in accuracy. This complexity may destroy the main essence of the proposed technique. That is why the technique is devised based on a heuristic approach rather than utilizing intense mathematical calculation. Simple approach like plotting a histogram is an easy way to determine a good  $\tau_{\min}$ .

### IV. RESULTS

We implemented the reduction algorithm in C++. Specifically,  $p$  was assumed to be zero in (7). We also required  $\phi_{\max} = 0$ , which guarantees no refills by the reduction algorithm and maintains the sparsity of the  $Y$  matrix. Therefore, only nodes with a degree less than or equal to three were candidates for elimination. By applying the methods on some small industrial circuits (Table I and Fig. 5) with less than 500

TABLE I  
SMALL- AND MEDIUM-SIZE INDUSTRIAL CIRCUITS

	Small industrial circuit (less than 100 nodes)				Medium industrial circuit (several hundred nodes)			
	Original	Reduced circuit for different $\tau_{min}$ (ps)			Original	Reduced circuit for different $\tau_{min}$ (ps)		
$\tau_{min}$ (ps)	-	0.002	20	200	-	80	180	192
Nodes	81	55	43	29	227	158	147	127
Total Elements	152	105	79	54	526	290	271	241
% Reduction of nodes	-	32	47	64	-	30.39	35.24	44.1
% Reduction of elements	-	31	48	64.5	-	44.87	48.47	54.2
% Error in Rise Time	-	0.02	0.8	0.7	-	0.09	3.1	13.7
% Error in Delay	-	0.36	0.36	11	-	0.34	0.97	7.5

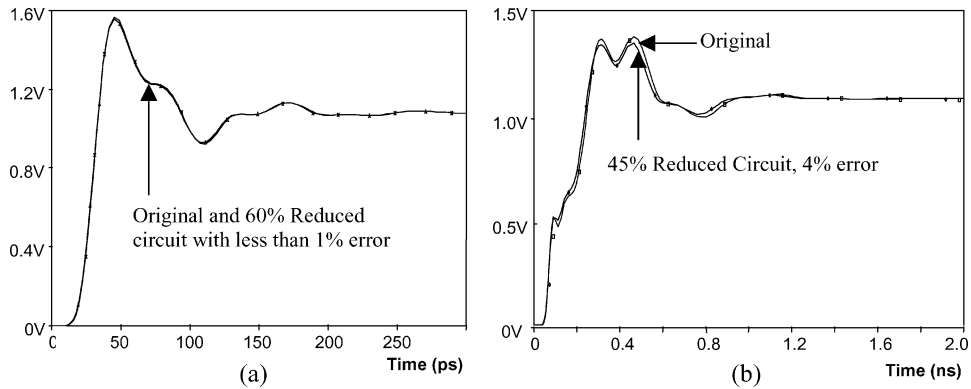


Fig. 5. Sample waveforms from (a) small and (b) medium industrial circuits.

TABLE II  
BALANCED AND UNIFORM H-TREE NETWORKS

	Original	Reduced circuit for different $\tau_{min}$ (ns)		
		0.1	0.5	1
$\tau_{min}$ (ns)	-	0.1	0.5	1
Nodes	2502	666	416	128
Total Elements (R: 1250, L: 1250, C: 1250)	3750	997	617	190
% Reduction of nodes	-	73.4	83.37	94.9
% Reduction of elements	-	73.4	83.54	94.9
% Error in Rise Time	-	0.00	0.01	0.03
% Error in Delay	-	0.00	0.00	0.00

nodes an average 50% reduction of circuit elements and nodes is obtained with less than 1% error in rise time and delay calculation. The range of driver impedances was comparable to the total impedance of the interconnect. If the allowable margin of error is around 3% an average reduction of 55% can be achieved by selecting a higher  $\tau_{min}$ . Higher error tolerance (around 5% and higher) will give even higher reduction in the range of 55% to 70%.

For symmetric and uniformly distributed *RLC* networks, this method will give very high reduction. For a balanced and uniform H-tree network, a reduction of about 90% is obtained with almost 0% error (Table II and Fig. 6). With this amount of reduction, a twenty-fold decrease in simulation time is obtained. Such a high reduction of the simulation time is due to the symmetric and uniform nature of the original and resultant reduced circuits.

The above small- and medium-sized circuits do not give clear idea of speed up (reduction of actual simulation time) due to nodes and elements reduction. With these objectives in mind the method is applied to an extracted clock-distribution network from a commercial

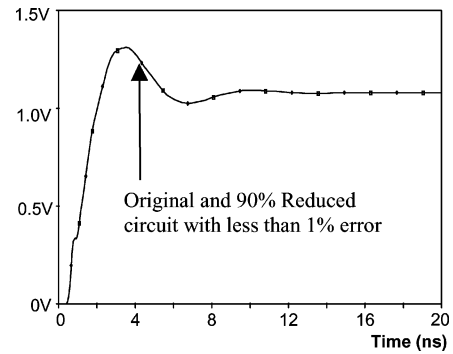


Fig. 6. Sample signals from balanced and uniform H-tree network.

high-performance microprocessor. The extracted circuit contained over 678 608 elements and more than 10 000 sinks. A histogram of the distribution of the nodal time constants of this circuit up to a maximum of 100 ps is shown in Fig. 7.

Based on this histogram, we made three choices of  $\tau_{min}$  for circuit reduction:  $\tau_{min} = 15$  ps,  $\tau_{min} = 25$  ps, and  $\tau_{min} = 35$  ps. The results of running RICE (v5) [4] on the original as well as the three reduced circuits are shown in Table III for an input rise time of 50 ps. A fourth-order approximation was computed at every sink. Clearly,  $\tau_{min} = 15$  ps offers the best choice, resulting in over three times decrease in analysis time with almost no loss in accuracy. These reductions in runtime are expected to be much higher when nonlinear elements are combined with the interconnects. Note that the reduced *RLC* circuit can be readily inserted in any simulator, such as SPICE or AS/X. As expected, choosing larger values of  $\tau_{min}$  resulted in a loss of accuracy. Besides improving the analysis

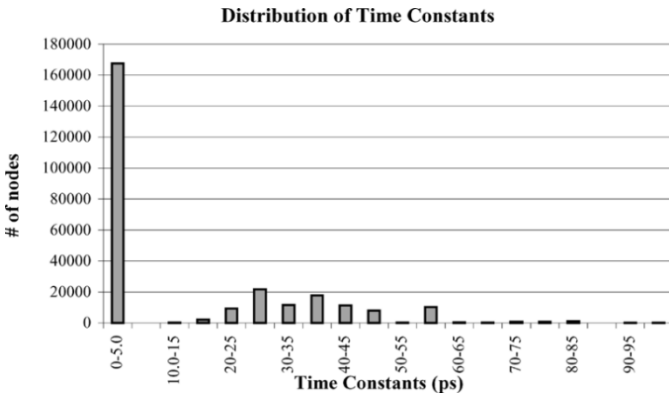


Fig. 7. Distribution of time-constants for different nodes of a clock-distribution network.

TABLE III  
REDUCED COMMERCIAL CIRCUIT STATISTICS

	Original	Reduced circuit for different $\tau_{min}$ (ps)		
		15	25	35
$\tau_{min}$ (ps)	-	15	25	35
Nodes	380K	148K	141K	113K
Total Elements (R: 368K, L: 72K, C: 239K)	679K	285K	277K	239K
% Reduction of nodes	-	60.8	62.8	70.1
% Reduction of elements	-	57.9	59.2	64.6
% Error in Rise Time	-	0.05	-1.80	-11.1
% Error in Delay	-	0.25	10.5	21.7
Speedup	-	3.23	3.26	3.54

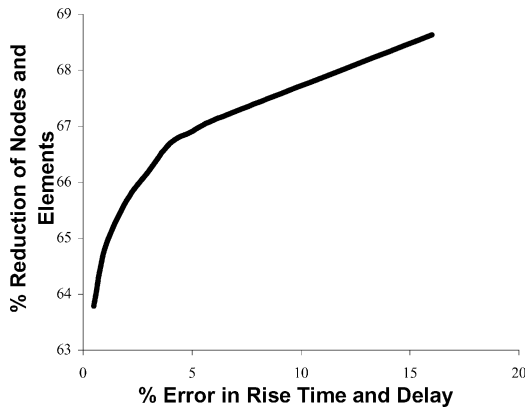


Fig. 8. % Error versus reduction.

time, the reduction algorithm also reduced the storage requirements by one-third.

The average reduction at different levels of error in delay and rise time calculation is shown in Fig. 8 from the reduction statistics of all the circuits presented above. It is observed that if the error is limited to 1%, an average reduction of 60% is obtained. If the error is allowed to go up to 3%, an average reduction of 66% is obtained. At 5% error, an average of 67% reduction can be achieved. It is observed from Fig. 8 that beyond a 67% reduction point, for a smaller gain in reduction, a very high error is introduced. This is because at these amounts of reduction, many critical circuit nodes and elements are thrown out of the netlist. Consequently, the circuit behavior changes sufficiently and very high error in performance evaluation occurs.

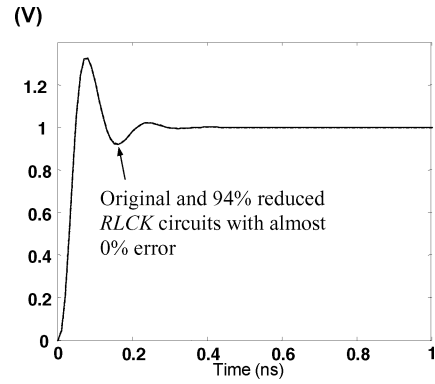


Fig. 9. Sample waveform from 16-bit coupled RLC bus with coupling-inductors.

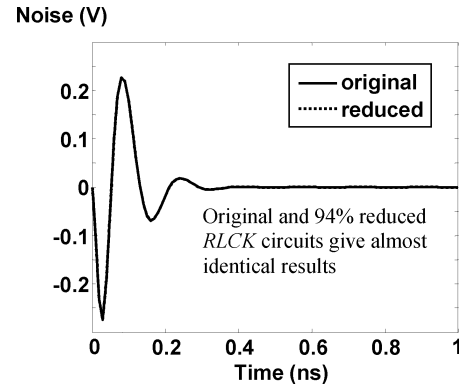


Fig. 10. Noise on a victim bit-line of the 16-bit coupled bus.

The RLCK node elimination scheme (see Appendix ) was applied to a 16-bit coupled bus with 18 000 elements (3200 R, 3200 L, 3200 C, 8400 K) including coupling-inductors and 9617 nodes. The circuit was reduced down by 94% with the new circuit having 189R, 205C, 189L, and 916K. The results show that the method reduces the number of coupling-inductors by as much as 89%. A sample output for a middle bit is shown in Fig. 9. The output signal for the reduced circuit is almost identical to the signal in original circuit with 18 000 elements. Error in rise-time and delay was almost 0%. The results for other nodes are also very accurate.

The results for crosstalk prediction are also very accurate, using the reduced circuit. Fig. 10 shows the noise on a quiet victim bit-line of the 16-bit coupled bus because of an aggressor bit-line switching. The noise curve given by the reduced circuit is almost identical to the noise curve given by the original circuit. The results are very similar at other nodes also. These accurate results on 94% reduced RLCK bus show that the proposed method is highly applicable to circuits with coupling-inductors. Not only a large 89% reduction in the number of coupling inductors is achieved, but the reduced circuit also gives signals that are almost identical to those produced by the large original netlist.

### V. CONCLUSION

This paper presented a realizable reduction method of RLC circuits (with coupling-inductors) by nodal elimination. This method is useful in analyzing and verifying large RLC networks. If MOR is not realizable, it produces a reduced mathematical model of transfer function or reduced state equations. Hence, all downstream circuit simulation and associated tests have to be modified to handle these reduced mathematical representations of the circuits. Since the proposed realizable reduction method is an RLCK-in to RLCK-out, all standard simulators can

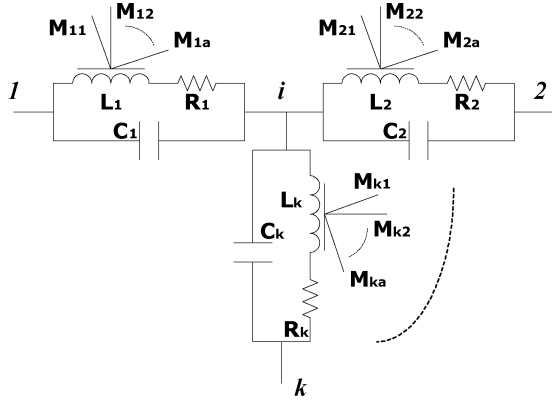


Fig. 11. General node in  $RLC$  circuit with coupling inductors.

handle the reduced circuits without any modification. Reduced netlist also guarantees faster simulation, lower memory and storage requirements. It is shown that for an average of 40% to 50% reduction of nodes and elements, the error in waveform shape calculation is less than 1%. If 3% error is allowed, a reduction of about 60% to 70% is obtained. A higher allowance of error results in an even higher reduction. Therefore, the user has the freedom to make the trade-off between accuracy and speed. Circuits with coupling-inductors are also reducible with the presented algorithm (see Appendix) and the results are very accurate for aggressor as well as victim nets.

#### APPENDIX RLCK NETLIST REDUCTION

The circuit-reduction scheme presented in Section II can be extended easily to include  $RLC$  circuits with coupling-inductors (which are represented by the letter  $K$ ). Fig. 11 shows the general node  $i$  of an  $RLC$  circuit with coupling-inductors. In Fig. 11, each admittance has been decomposed into the generalized  $RLC$  branch form of Fig. 2 and coupling-inductors are added to the  $RL$  branch.

The  $i$ th row of (1) is now given by:

$$Y_i V_i + sC_i V_i - y_1 \left( V_1 - \sum_{x=1}^{1a} sM_{1,x} I_{1,x} \right) - y_2 \left( V_2 - \sum_{x=1}^{2a} sM_{2,x} I_{2,x} \right) - \dots$$

$$- y_k \left( V_k - \sum_{x=1}^{ka} sM_{k,x} I_{k,x} \right) - sC_1 V_1 - sC_2 V_2 - \dots - sC_k V_k = 0$$

$$\text{where } Y_i = \sum_{j=1}^k y_j \text{ and } C_i = \sum_{j=1}^k c_j. \quad (14)$$

Unlike (2),  $y$  here represents the  $RL$  branch only and  $c$  represents the capacitive branch of the generalized admittance. Coupling-inductors are represented by  $M_{j,x}$  ( $j = 1 \dots k$ ;  $x = 1, 2, \dots, j_a$ ) in (14) and they induce a voltage drop on the  $RL$  branch proportional to  $sM_{j,x} I_{j,x}$ , where  $I_{j,x}$  is the current passing through an inductor  $L_x$ , which is coupled with the inductor  $L_j$ . For generalized admittance  $j$ , there are  $j_a$  coupling-inductors. In order to eliminate  $V_i$  from the system, we solve for  $V_i$  using (14) and obtain (15) found at the bottom of the page.

Equation (16) describes the row corresponding to the first neighbor of node  $i$  where  $k_1$  is the number of nodes connected to node 1

$$Y_1 V_1 + sC_1 V_1 - y_1 \left( V_i + \sum_{x=1}^{1a} sM_{1,x} I_{1,x} \right) - y_2 \left( V_2 - \sum_{x=1}^{2a} sM_{2,x} I_{2,x} \right) - \dots - y_{k_1} \left( V_{k_1} - \sum_{x=1}^{k_1 a} sM_{k_1,x} I_{k_1,x} \right) - sC_1 V_1 - sC_2 V_2 - \dots - sC_{k_1} V_{k_1} = 0. \quad (16)$$

To eliminate node  $i$ , we substitute value of  $V_i$  as from (15) in (16) and obtain

$$\left( \hat{Y}_1 + s\hat{C}_1 + y_1 + sC_1 - \frac{(y_1 + sC_1)^2}{Y_i + sC_i} \right) V_1 - y_1 \sum_{x=1}^{1a} sM_{1,x} I_{1,x} - \frac{\sum_{j=2}^k (y_1 + sC_1)(y_j + sC_j) V_j}{Y_i + sC_i} + \frac{\sum_{j=1}^k (y_1 + sC_1) y_j \sum_{x=1}^{j_a} sM_{j,x} I_{j,x}}{Y_i + sC_i} - \sum_{\substack{r=1 \\ r \neq i}}^{k_1} y_r \left( V_r - \sum_{x=1}^{r_a} sM_{r,x} I_{r,x} \right) - \sum_{\substack{r=1 \\ r \neq i}}^{k_1} sC_r V_r = 0$$

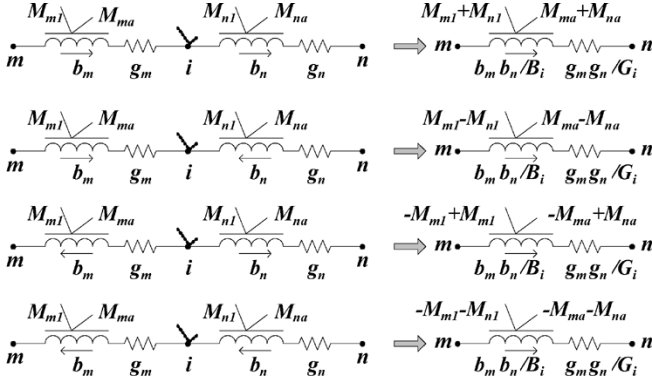
where  $\hat{Y}_1 = \sum_{r=1, r \neq i}^{k_1} y_r$  and  $\hat{C}_1 = \sum_{r=1, r \neq i}^{k_1} c_r$  (17)

where  $\hat{Y}_1$  is the sum of all admittances ( $RL$ ) from node 1 except to node

$$V_i = \frac{y_1 \left( V_1 - \sum_{x=1}^{1a} sM_{1,x} I_{1,x} \right) + y_2 \left( V_2 - \sum_{x=1}^{2a} sM_{2,x} I_{2,x} \right) + \dots + y_k \left( V_k - \sum_{x=1}^{ka} sM_{k,x} I_{k,x} \right) + sC_1 V_1 + sC_2 V_2 + \dots + sC_k V_k}{Y_i + sC_i} \quad (15)$$

$$\left( \hat{Y}_1 + s\hat{C}_1 + \frac{\sum_{j=2}^k (y_1 + sC_1)(y_j + sC_j)}{Y_i + sC_i} \right) V_1 - \frac{\sum_{j=2}^k (y_1 + sC_1)(y_j + sC_j) V_j}{Y_i + sC_i} - \frac{\sum_{j=2}^k (y_j + sC_j) y_1 \sum_{x=1}^{1a} sM_{1,x} I_{1,x} + \sum_{j=2}^k (y_1 + sC_1) y_j \sum_{x=1}^{j_a} sM_{j,x} I_{j,x}}{Y_i + sC_i} - \sum_{\substack{r=1 \\ r \neq i}}^{k_1} y_r \left( V_r - \sum_{x=1}^{r_a} sM_{r,x} I_{r,x} \right) - \sum_{\substack{r=1 \\ r \neq i}}^{k_1} sC_r V_r = 0 \quad (18)$$

$$\begin{aligned}
 & (y_{mn} + sc_{mn}) + y_{mn} \sum_{x=1}^{m n_a} sM_{mn,x} I_{mn,x} \\
 & \equiv \frac{(y_m + sc_m)(y_n + sc_n) - (y_n + sc_n)y_m \sum_{x=1}^{m_a} sM_{m,x} I_{m,x} + (y_m + sc_m)y_n \sum_{x=1}^{n_a} sM_{n,x} I_{n,x}}{Y_i + sC_i}
 \end{aligned} \tag{19}$$



Notes:

1. Arrows on inductors indicate assumed current flow direction .
2. For RC, LC, and C branches, rules are same as those in Figure 3.

Fig. 12. Rules for eliminating node  $i$  in  $RLCK$  circuit.

$i$  and  $\hat{C}_1$  is the sum of all capacitances from node 1 except to node  $i$ . Equation (17) can be simplified as (18) found at the bottom of the previous page. Note that this is equivalent to adding  $k - 1$  new elements between node 1 and the  $k - 1$  former neighbors of node  $i$ . Equation (18) is very similar to (5), except that the admittances are split between the  $RL$  and the  $C$  branches, and there are additional terms representing coupling-inductors. Specifically, for any two neighbors of node  $i$ , say  $m$  and  $n$ , the elimination of node  $i$  results in the addition of a new generalized admittance between nodes  $m$  and  $n$  whose equivalence is given by (19) found at the top of the page, where it is assumed that current flows from  $m$  to  $i$  and from  $n$  to  $i$  before eliminating node  $i$  and it flows from  $m$  to  $n$  after eliminating node  $i$ . For low-frequency approximation,  $s^2$  terms in (19) are ignored. Thus, when node  $i$  is eliminated, the coupling-inductors present on the  $RL$  branch between node  $m$  and node  $i$  are copied over to the new  $RL$  branch between nodes  $m$  and  $n$ . Similarly, all the coupling-inductors present on the  $RL$  branch between node  $i$  and node  $n$  are also copied over to the new  $RL$  branch between nodes  $m$  and  $n$ . The  $RLC$  of the new admittance between nodes  $m$  and  $n$  is obtained the same way as for circuits without coupling-inductors. Thus, for circuits without coupling inductors, the elimination procedure falls back to the scheme presented in Section II. Fig. 12 shows the rules for eliminating node  $i$ , for a general  $RLC$  circuit with coupling-inductors. Again, this type of nodal elimination preserves nodal voltages and equivalent branch admittance between nodes, including coupling-inductors. Although the branch currents are not preserved explicitly, they are preserved implicitly because they only depend on nodal voltages and equivalent branch admittances, which are preserved.

#### ACKNOWLEDGMENT

The authors thank C. Kashyap and B. Krauter from the IBM Corporation, Austin, TX, for their valuable feedback and help for this research paper.

#### REFERENCES

- [1] A. Deutsch, P. W. Coteus, G. V. Kopcsay, H. H. Smith, C. W. Surovic, B. L. Krauter, D. C. Edelstein, and P. J. Restle, "On-chip wiring design challenges for gigahertz operation," *Proc. IEEE*, vol. 89, no. 4, Apr. 2001.
- [2] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of merit to characterize the importance of on-chip inductance," *IEEE Trans. VLSI Syst.*, vol. 7, no. 4, pp. 442–449, Dec. 1999.
- [3] *SPIRAL 3D Spiral Inductor Synthesis Tool*, OEA International, Inc., 2002.
- [4] C. L. Ratzlaff, N. Gopal, and L. T. Pillage, "RICE: Rapid interconnect circuit evaluator," in *Proc. Design Automation Conf.*, 1991, pp. 555–560.
- [5] L. T. Pillage and R. A. Rohrer, "Asymptotic waveform evaluation for timing analysis," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 9, no. 4, pp. 352–366, Apr. 1990.
- [6] Feldmann and R. W. Freund, "Efficient linear circuit analysis by Pade approximation via the Lanczos process," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 14, no. 5, pp. 639–649, May 1995.
- [7] M. Silveria, M. Kamon, I. Elfadel, and J. White, "A coordinate transformed Arnoldi algorithm for generating guaranteed stable reduced-order models of arbitrary  $RLC$  circuits," in *Proc. Int. Conf. Computer-Aided Design*, 1996, pp. 288–294.
- [8] A. Odabasioglu, M. Celik, and L. T. Pileggi, "PRIMA: Passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. Computer-Aided Design Integr. Circuits Syst.*, vol. 17, no. 8, pp. 645–654, Aug. 1998.
- [9] A. Devgan and P. R. O'Brien, "Realizable reduction for  $RC$  interconnect circuits," in *Proc. Int. Conf. Computer-Aided Design*, 1999, pp. 204–207.
- [10] P. J. H. Elias and N. P. van der Meijs, "Extracting circuit models for large  $RC$  interconnections that are accurate up to a predefined signal frequency," in *Proc. Design Automation Conf.*, 1996, pp. 764–769.
- [11] A. J. van Genderen and N. P. van der Meijs, "Extracting simple but accurate  $RC$  models for VLSI interconnect," in *Proc. Int. Symp. Circuits Syst.*, 1988, pp. 2351–2354.
- [12] B. N. Sheehan, "TICER: Realizable reduction of extracted  $RC$  circuits," in *Proc. Int. Conf. Computer-Aided Design*, 1999, pp. 200–203.
- [13] P. J. H. Elias and N. P. van der Meijs, "Efficient moments extraction of large inductively coupled interconnection networks," in *Proc. Int. Symp. Circuits Syst.*, 1996, pp. IV 540–IV 543.