Realizable Reduction for Magnetic and Electric Coupled RLKC Interconnects

Rong Jiang and Charlie Chung-Ping Chen
Electrical and Computer Engineering, University of Wisconsin-Madison

ABSTRACT

Due to the rising clock frequency and integration density, inductance effects quickly become another signal integrity concern besides capacitance coupling. Beside self inductance, the size explosion of mutual inductance elements creates tremendous challenges for timing, noise, and even power integrity analysis. In this paper, we present a novel simultaneous $RLKC$ reduction algorithm which simultaneously reduces all elements by several novel reduction primitives. Most of our reduction primitives are compatible with general realizable RC reduction algorithms such as [8] and hence can coexist or be embedded into general RC reduction softwares. The reduced circuits are also realizable $RLKC$ elements and hence are compatible with general circuit simulators. Experimental results show that our algorithm is extremely efficient and accurate. With linear runtime and memory consumption, our algorithm takes only 0.3 seconds to reduce a circuit with over 10,000 passive elements while maintains less than 3% error and 50% reduction ratio.

1. INTRODUCTION

As VLSI technology advances into Ultra Deep Sub-Micron (UDSM) and multi-giga hertz clock frequency, interconnect has become an increasingly important in determining system performance and reliability. With the introduction of copper and low-k dielectric, the inductive effects gradually emerges as another major concerns for signal integrity analysis besides capacitance coupling effects. Due to the long range inductive coupling effects, the size of mutual inductance elements grows quadratic in proportional with interconnect elements and create huge dense inductance matrix. Dense matrix is special hard to sparse matrix solvers which are used by SPICE. As a result, model order reduction techniques are essential to speed up the analysis turn around time.

To deal with this effects, several moment matching based model reduction algorithms such as AWE [6], PVL [2], PRIMA [5], [3], [7], and [9] have been proposed. However, the reduced model generated from those algorithms may not be realizable by passive elements from the frequency domain information. Under these circumstances, realizable reduction algorithms have been proposed, such as [1], [4] and TICER [8]. Based on node elimination, TICER properly removes circuit nodes with insignificant or out-of-bound time constants without much accuracy penalty. Unfortunately, TICER doesn't apply when inductive components exist. Although the self inductance has been consider in [4], the mutual inductance terms are not considered. Furthermore, [4] doesn't take capacitance into consideration. Note that the current return path is closed related to capacitance return paths since the impedance of capacitance is much lower for higher frequency. As a result, none of the exiting reduction algorithm can fulfill the general realizable RLKC reduction tasks.

In this paper, we propose to simultaneously reduce $RLKC$ elements by several novel reduction methods such as serial, parallel, RC, and LKC reductions. We iteratively search the circuits and apply the reduction rules when it is applicable. One of the advantages of our reduction methods is that the reduced model is also passive linear $RLKC$ elements and hence is compatible with most of the circuit simulators. Experimental results show that our algorithm is extremely efficient and accurate. With linear runtime and memory consumption, our algorithm takes only 0.3 seconds to reduce a circuit with over 10,000 passive elements while maintains less than 3% error and 50% reduction ratio.

The remainder of this paper is organized as follows. In Section 2, we present the overview of our reduction algorithms. In section 3, the aforesaid methods for combining RL branches and splitting capacitances are introduced in details. In Section 4, numerical experiments are given. In Section 5, we make some conclusion remarks.

2. OVERVIEW OF OUR REDUCTION ALGORITHMS

2.1 Reduce Serial Inductances

In this section, we will briefly overview our five $RLKC$ methods, Method 1-5. Based on these methods, a high-level $RLKC$ reduction algorithm is presented.

For VLSI interconnects, extracted $RLKC$ circuits generally include resistance branches, capacitance branches and RL branches. In Figure 1, we show the parasitics for a five-bit bus with the middle bit as ground. Our first reduction method, Method 1, is for combining serial RL branches can be applied significantly reduce the circuits. This process is

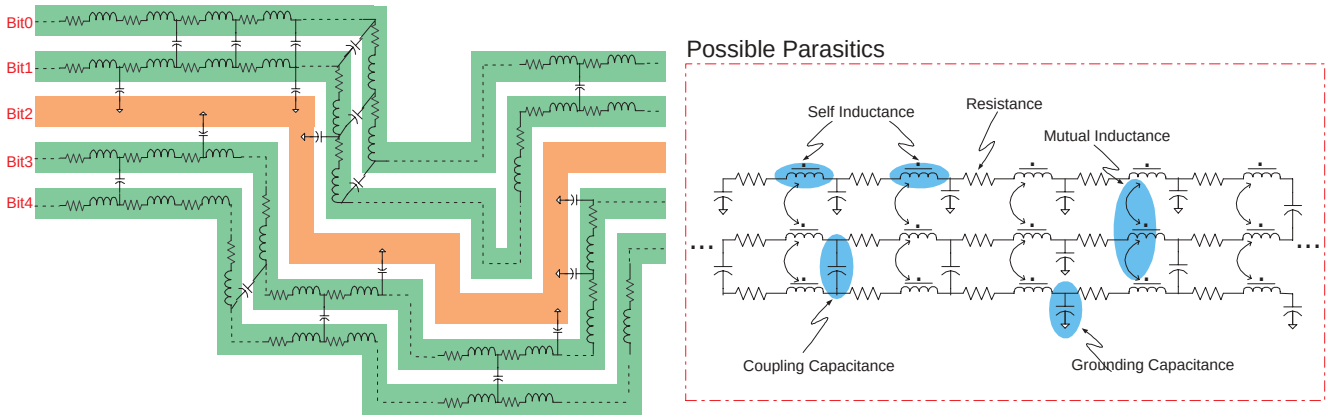

Figure 1: A Five-bit Bus and Parasitics Including Mutual Inductances and Coupling Capacitances.

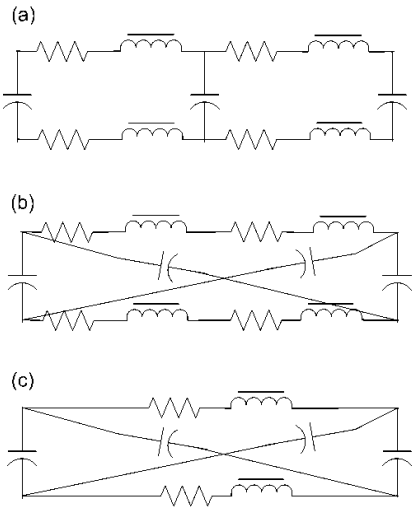

Figure 2: Serial RLKC reduction (a) Original. (b) RC reduction (c) LK reduction .

illustrated in Figure 2.

Method 2 is for combining parallel inductances. After applying these two methods, most nodes in $RLKC$ interconnect circuits will have two RL branches and capacitances on them. We show in Figure 3 that by swapping the position of resistances and inductances in attached two RL branches properly, RC nodes in structure A and B or LC nodes in structure C and D are obtained. RC nodes can be classified into quick nodes, slow nodes and normal nodes according to their time constants [8]. Quick nodes and slow nodes are those with extreme RC time constants and hence can be eliminated without changing the circuit characteristics too much. For the LC nodes, we propose Method 3, 4and 5 to split the attached capacitances to neighboring nodes. After splitting, some parallel capacitances can be combined with each other, and two serial inductances are obtained. Then Method 1 for combining serial inductances can be applied.

We present our high-level reduction algorithm in Table 1.

In the following section, we will present methods in Table 1 in details. And we also show that our methods for LC nodes reduction is actually a process to eliminate insignificant LC time constants.

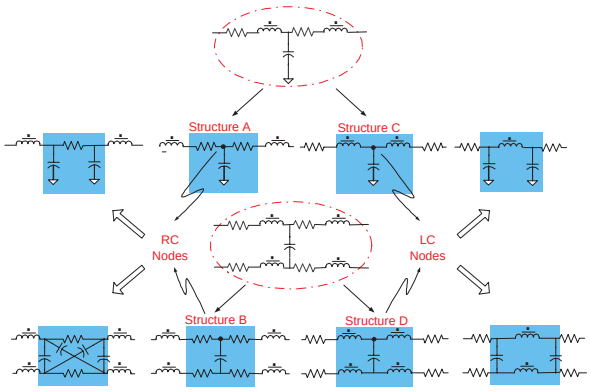

Figure 3: Swapping Elements to Obtain RC or LC Nodes.

Algorithm: Realizable RLKC Reduction
Begin:
  Case 1: Two RL branches in serial
    Apply RC Reduction if applicable;
    Apply Method 1 to combine serial RL branches;
  Case 2: Two RL branches in parallel
    Apply Method 2 to combine parallel RL branches;
  Case 3: Capacitances $\{C_i\}_{i=1}^{n}$between two RL branches
    Let L_1, L_2inductances in RL branches, respectively;
    Swap elements in RL branches to obtain LC node;
    Split capacitances by using Method 3, 4 or 5;
    Apply Method 1 to reduce serial inductances;
    Apply RC Reduction if applicable;
End

Table 1: Realizable RLKC Reduction Algorithm

3. REALIZABLE REDUCTION METHODS

In this section, we will first discuss how to reduce serial

and parallel RL branches. Then three methods are presented for splitting capacitances on an LC node.

On our way to realizable $RLKC$ circuits reduction, self and mutual inductance seem to be the troublemakers. In this subsection, we focus on a circuit totally comprising of inductances. We will show how to combine serial inductances and update related mutual inductances.

Given a circuit that has n branches and each branch contains an inductance. The inductance in the i^{th} branch is $L_{i,i}$. $L_{i,i}$ is coupled with inductances in other branches. The mutual inductance between $L_{i,i}$ and $L_{j,j}$, the inductance in the j^{th} branch, is $L_{i,j}$. From elementary electrical circuit theory, we know that $L_{i,j} = L_{j,i}$. Let I_i the current flowing through the i^{th} branch and V_i the branch voltage drop in the associated direction of I_i. The following equation gives the relation of branch currents and voltage drops of this circuit.

$$\begin{bmatrix} \dot{i}_1 \\ \dot{i}_2 \\ \vdots \\ \dot{i}_n \end{bmatrix} = \begin{bmatrix} L_{11} & L_{12} & \cdots & L_{1n} \\ L_{21} & L_{22} & \cdots & L_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ L_{n1} & L_{n2} & \cdots & L_{nn} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

3.1 Serial Reduction

If branch 1 and 2 are serially connected, $i_1 = i_2 = i_{join}$. Let $v_{join} = v_1 + v_2$, the above equation can be rewritten as the following equation by summing up row 1 and row 2, column 1 and column 2. Let $L_{join} = L_1 + 2L_{1,2} + L_2$ and $L_{join,j} = L_{j,join} = L_{1,j} + L_{2,j}$. we have:

$$\begin{bmatrix} \dot{i}_{join} \\ \vdots \\ \dot{i}_j \\ \vdots \\ \dot{i}_n \end{bmatrix} = \begin{bmatrix} L_{11}+L_{12}+2L_{12} & \cdots & \cdots & L_{1n}+L_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ L_{j1}+L_{j2} & \cdots & \cdots & L_{jn} \\ \vdots & \vdots & \vdots & \vdots \\ L_{n1}+L_{n2} & \cdots & \cdots & L_{nn} \end{bmatrix} \begin{bmatrix} v_{join} \\ \vdots \\ v_j \\ \vdots \\ v_n \end{bmatrix}$$

which is simply summing up rows 1 and 2, and columns 1 and 2 together, and then we can find the new inductance matrix.

The above derivation has its physical meaning which can be translated into a method to combine two serially connected RL branches. See Figure 4.

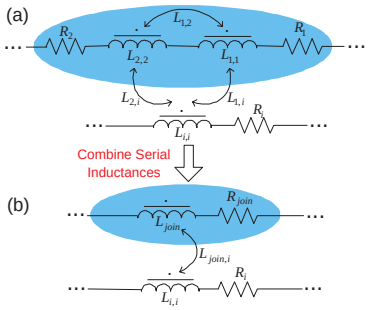

Figure 4: Combine Serial RL Branches.

We summarize the method as follows:

METHOD 1. *(Combine Serial RL Branches) For two serially connected RL branches with inductances, $L_{i,i}$ and $L_{j,j}$, we can combine them together and denote the combined inductance as L_{join}. Then $L_{join} = L_{i,i} + L_{j,j} + 2L_{i,j}$, $R_{join} = R_1 + R_2$ and the mutual inductances between combined inductance L_{join} and the rest inductances are $L_{join,j} = L_{1,j} + L_{2,j}$.*

By applying the above method, nodes between two serial RL branches can be reduced. Next subsection will introduce method for combining parallel RL branches.

3.2 Parallel Reduction

In an $RLKC$ interconnect circuit, some RL branches may in parallel. For simplicity, we first consider two RL branches in parallel. See Figure 5.

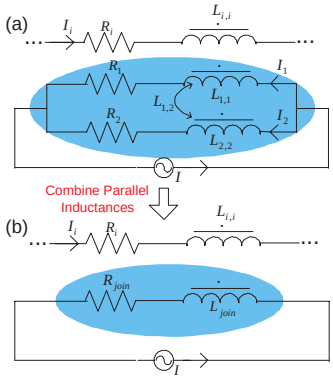

Figure 5: Combine Parallel RL Branches.

If the two parallel RL branches in circuit (a) can be combined as shown in circuit (b), for maintaining the circuit characteristic, the port voltages must be same when applying same current sources. For circuit (a), the port voltage can be expressed by:

$$V_a = \frac{R_1R_2 + s(R_1L_{2,2} + R_2L_{1,1}) + s^2(L_{1,1}L_{2,2} - L_{1,2}^2)}{R_1 + R_2 + s(L_{1,1} + L_{2,2} - 2L_{1,2})} I + \frac{s(R_1L_{2,i} + R_2L_{1,i}) + s^2[(L_{2,2} - L_{1,2})L_{1,i} + (L_{1,1} - L_{1,2})L_{2,i}]}{R_1 + R_2 + s(L_{1,1} + L_{2,2} - 2L_{1,2})} I_i.$$

Similarly for circuit (b), the port voltage is:

$$V_b = (R_{join} + sL_{join})I + sL_{join,i}I_i$$

Let $s = j\omega$, we can proof that when the inductive components of these RL branches are much smaller than R_1 and R_2,

$$V_a \approx (\widehat{R}_{join} + s\widehat{L}_{join})I + s\widehat{L}join,iI_i,$$

where

$$\begin{aligned} \widehat{R}_{join} &\approx \frac{R_1R_2}{R_1+R_2} \\ \widehat{L}_{join} &\approx \frac{R_1^2L_{2,2} + R_2^2L_{1,1} + 2R_1R_2L_{1,2}}{(R_1+R_2)^2} \\ \widehat{L}_{join,i} &\approx \frac{R_1L_{2,i} + R_2L_{1,i}}{R_1+R_2} \end{aligned}$$

By matching the coefficients of I and I_i, we propose the method for combining two parallel RL branches:

METHOD 2. *(Combine Parallel RL Branches) Given two RL branches as shown in Figure 5.(a), if $R_1 + R_2 >> \omega(L_{1,1}+L_{2,2}-2L_{1,2})$, these two branches can be combined as shown in circuit 5.(b). The combined resistance and inductance are $R_{join} = \frac{R_1R_2}{R_1+R_2}$ and $L_{join} = \frac{R_1^2L_{2,2}+R_2^2L_{1,1}+2R_1R_2L_{1,2}}{(R_1+R_2)^2}$. The mutual inductance between L_{join} and $L_{i,i}$ is $\frac{R_1L_{2,i}+R_2L1,i}{R_1+R_2}$.*

The above method for combining two RL branches can be easily extended to iteratively combine multiple parallel connected RL branches. After applying Method 1 and 2,

most nodes in $RLKC$ interconnect circuits will have two RL branches and some capacitances on them. In the next subsection, we will show how to split attached capacitances to outside nodes.

3.3 General RLKC Reduction

As we can see from the proceeding derivation, grounding and coupling capacitances prevent the use of Method 1 to combine serially connected inductances. See Figure 6. But

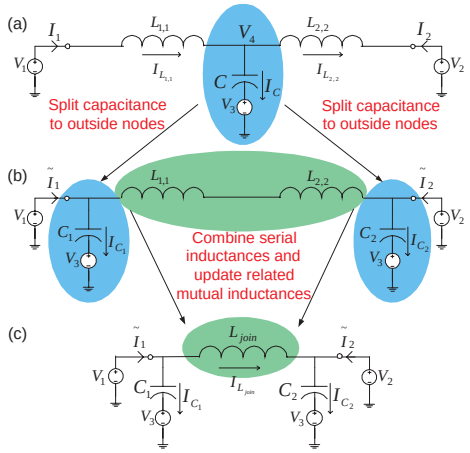

Figure 6: Split Capacitances.

if the capacitance in circuit (a) can be split to outside nodes as shown in circuit (b), Method 1 then can be applied to reduce circuit (b) to circuit (c).

Now assume port voltage sources V_1, $V_2$and $V_3$are independent. And in order to make discussion here suitable for both grounding and coupling capacitances, $V_3$is not set to zero. For maintaining the entire circuit characteristic, the port currents flowing into circuit (a) and circuit (b) cannot be changed before and after transformation, i.e. $I_1 = \widetilde{I}_1$and $I_2 = \widetilde{I}_2$. And also notice that the currents flow though inductances will be different in circuit (a) and (b), which will affect other circuit portions because of mutual inductance effect. For example, if branch j contains an inductance $L_{j,j}$which has mutual inductances $L_{1,j}$, $L_{2,j}$with $L_{1,1}$and $L_{2,2}$in circuit (a), respectively. The mutual inductance effect caused by $L_{1,1}$and $L_{2,2}$on branch j is $V_{before} = sL_{j,1}I_{L_{1,1}} + sL_{j,2}I_{L_{2,2}}$. If circuit (a) is changed to circuit (c), the mutual inductance effect turns to be $V_{after} = sL_{j,join}I_{join}$. So in order to preserve the electrical characteristic of the entire circuit, we must insure that $V_{before} \approx V_{after}$.

We can derive the following relations between port currents and voltages of circuit (a) in Figure 6:

$$I_1 = \frac{(1+K_1)V_1 - V_2 - K_1V_3 - \sum_{i=3}^{n} s[(1+K_1)L_{1,i} + L_{2,i}]I_{L_{i,i}}}{s[\overline{L}_{join} + K_1(L_{1,1}+L_{1,2})]},$$

$$I_2 = \frac{(1+K_2)V_2 - V_1 - K_2V_3 + \sum_{i=3}^{n} s[L_{1,i} + (1+K_2)L_{2,i}]I_{L_{i,i}}}{s[\overline{L}_{join} + K_2(L_{1,2}+L_{2,2})]},$$

where

$$K_1 = \frac{s^2(L_{1,2}+L_{2,2})C}{1-s^2L_{1,2}C},$$

$$K_2 = \frac{s^2(L_{1,1}+L_{1,2})C}{1-s^2L_{1,2}C},$$

$$\overline{L}_{join} = L_{1,1} + 2L_{1,2} + L_{2,2}.$$

Similarly, relations of port currents with port voltages of circuit (c) in Figure 6 are given by:

$$\widetilde{I}_1 = \frac{(1+s^2L_{join}C_1)V_1 - V_2 - s^2L_{join}C_1V_3 - \sum_{i=3}^{n} sL_{join,i}I_{L_{i,i}}}{sL_{join}},$$

$$\widetilde{I}_2 = \frac{(1+s^2L_{join}C_2)V_2 - V_1 - s^2L_{join}C_2V_3 + \sum_{i=3}^{n} sL_{join,i}I_{L_{i,i}}}{sL_{join}}.$$

Denote $\overline{L}_{join} + K_1(L_{1,1}+L_{1,2}) = \overline{L}_{join} + K_2(L_{2,1}+L_{2,2})$ as $\widehat{L}_{join}$. From above equations, we can see circuit (a) can be transformed into circuit (c) if the following conditions are satisfied: (1) $L_{join,i} = (1+K_1)L_{1,i} + L_{2,i}$; (2) $L_{join,i} = L_{1,i} + (1+K_2)L_{2,i}$; (3) $L_{join} = \widehat{L}_{join}$; (4) $s^2C_1L_{join} = K_1$ and $s^2C_2L_{join} = K_2$. Apparently, condition (1) and (2) cannot be satisfied simultaneously. But if $K_1, K_2 \ll 1$, that is when $s^2(2L_{1,2}+L_{2,2})C \ll 1$and $s^2(2L_{2,1}+L_{1,1})C \ll 1$, i.e. $s^2(L_{1,1}+2L_{1,2}+L_{2,2})C \ll 1$, we obtain $L_{join,i} \approx L_{1,i}+L_{2,i}$and $L_{join} \approx \overline{L}_{join} = L_{1,1}+2L_{1,2}+L_{2,2}$. Because $s^2(L_{1,1}+2L_{1,2}+L_{2,2})C \ll 1$implies $s^2L_{1,2}C \ll 1$, thus from condition (4), we get $C_1 \approx \frac{L_{1,2}+L_{2,2}}{L_{join}}C$and $C_2 \approx \frac{L_{1,1}+L_{1,2}}{L_{join}}C$. Because $C_1 + C_2 = C$, the total capacitance is kept unchanged before and after transformation. Also we can proof that if $s^2(L_{1,1}+2L_{1,2}+L_{2,2})C \ll 1$, $V_{before} \approx V_{after}$is satisfied. We will show how the above derivation can be applied to split grounding and coupling capacitances, respectively.

If the capacitance in Figure 6 is grounding, i.e. $V_3 = 0$, the above derivation can be applied directly. Method for splitting grounding capacitances is summarized as follows:

METHOD 3. *(Split Grounding Capacitances) If* $s^2(L_{1,1}+2L_{1,2}+L_{2,2})C \ll 1$, *the grounding capacitance in circuit (a) in Figure 6 can be split to neighboring nodes as shown in circuit (c). The two split capacitances are:* $C_1 \approx \frac{L_{1,2}+L_{2,2}}{L_{join}}C$ *and* $C_2 \approx \frac{L_{1,1}+L_{1,2}}{L_{join}}C$ *respectively.*

Figure 7 demonstrates how to handle coupled capacitances RL branches.

Based on the proceeding derivation, we know that circuit (a) in Figure 7 can be changed into circuit (b) if $s^2(L_{1,1}+2L_{1,2}+L_{2,2})C \ll 1$. The combined inductance in circuit (b) is $L_{join1} = L_{1,1}+2L_{1,2}+L_{2,2}$and capacitances are $\widetilde{C}_1 \approx \frac{L_{1,2}+L_{2,2}}{L_{join1}}C$and $\widetilde{C}_2 \approx \frac{L_{1,1}+L_{1,2}}{L_{join1}}C$. For the two capacitances $\widetilde{C}_1$and $\widetilde{C}_1$on node 2, we know that if condition $s^2(L_{3,3}+2L_{3,4}+L_{4,4})\widetilde{C}_1 \ll 1$and $s^2(L_{3,3}+2L_{3,4}+L_{4,4})\widetilde{C}_2 \ll 1$hold, i.e. $s^2(L_{3,3}+2L_{3,4}+L_{4,4})C \ll 1$, we can split these two capacitances individually. And then circuit (c) is obtained. Hence method for splitting coupling capacitances can be stated as follows:

METHOD 4. *(Split Coupling Capacitances) circuit (a) in Figure 7 can be changed into circuit (c), if the following two conditions hold:* $s^2(L_{1,1}+2L_{1,2}+L_{2,2})C \ll 1$*and* $s^2(L_{3,3}+2L_{3,4}+L_{4,4})C \ll 1$. *The new capacitances in circuit (c) are* $C_1 = \frac{L_{3,4}+L_{4,4}}{L_{join2}}\frac{L_{1,2}+L_{2,2}}{L_{join1}}C$, $C_2 = \frac{L_{3,3}+L_{3,4}}{L_{join2}}\frac{L_{1,1}+L_{1,2}}{L_{join1}}C$, $C_3 = \frac{L_{3,3}+L_{3,4}}{L_{join2}}\frac{L_{1,2}+L_{2,2}}{L_{join1}}C$ *and* $C_4 = \frac{L_{3,4}+L_{4,4}}{L_{join2}}\frac{L_{1,1}+L_{1,2}}{L_{join1}}C$.

In some cases, when $s^2(L_{1,1}+2L_{1,2}+L_{2,2})C$and $s^2(L_{3,3}+2L_{3,4}+L_{4,4})C$is small enough, we can move $C_2$and $C_3$to outside nodes, and circuit (d) can be obtained with capacitances $C_1 = 1/2(\frac{L_{1,2}+L_{2,2}}{L_{join1}} + \frac{L_{3,4}+L_{4,4}}{L_{join2}})C$and $C_4 = 1/2(\frac{L_{1,1}+L_{1,2}}{L_{join1}} + \frac{L_{3,3}+L_{3,4}}{L_{join2}})C$.

Now we have methods for splitting grounding or coupling capacitance, respectively. But in practice, intercon-

Circuit	Before Reduction		After Reduction		Runtime	Reduction Ratio		Memory
	♯ Node	♯ Element	♯ Node	♯ Element	(s)	Nodes	Elements	(MB)
c43	43	243	24	89	0.011	0.56	0.37	0.01
c146	146	2752	78	881	0.022	0.53	0.32	1
c397	397	19654	204	5490	0.231	0.51	0.28	5
c626	626	49422	320	13522	0.921	0.51	0.27	12
c794	794	79304	400	20786	1.443	0.50	0.26	19
c1046	1046	137377	528	35756	3.255	0.50	0.26	33
c1496	1496	280627	750	71617	7.822	0.50	0.25	68

Table 2: Some experimental results

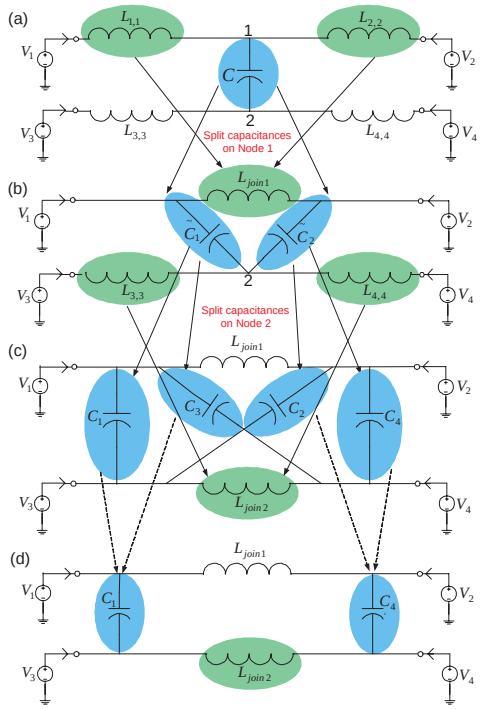

Figure 7: Split Coupling Capacitances.

nect circuits may have more complicated structures than those in Figure 6 and 7. For example, nodes between two RL branches may have both coupling and grounding capacitances attached. Hence a more general method to handle capacitances is required. Notice that Method 3 is obtained by first splitting the capacitance on node 1 and then splitting the two capacitances on node 2, respectively. Hence splitting capacitances on node 1 and node 2 are independent which means we can handle nodes one by one. The general method for splitting capacitances on an LC node is presented as follows:

METHOD 5. *(Split Capacitances on a LC node) Given an LC node with two inductances* $L_{1,1}$, $L_{2,2}$ *and* n *capacitances* $\{C_i\}_{i=1}^{n}$. *Let* $L_{join} = L_{1,1} + 2L_{1,2} + L_{2,2}$. *If* $s^2(L_{1,1} + 2L_{1,2}+L_{2,2})C_i \ll 1$ *holds, capacitance* C_i *can be split to outside nods with new capacitances* $C_{i,1} = \frac{L_{1,2}+L_{2,2}}{L_{join}}C_i$ *and* $C_{i,2} = \frac{L_{1,1}+L_{1,2}}{L_{join}}C_i$. *If all capacitances can be split, the obtained two serial inductances,* $L_{1,1}$ *and* $L_{2,2}$, *can be combined to* L_{join}.

Based on those proposed methods, especially Method 1 and 5, LC nodes can be reduced efficiently. In the next subsection, we will show that the reduced LC nodes are actually those with insignificant time constants. And our reduction algorithm preserves the damping factor of the circuit under reduction.

3.4 Preservation of Damping Factor and Time Constants

As we can see from the proceeding discussion, our algorithm reduces an LC node when $s^2(L_1 + 2L_{1,2} + L_2)C$is very small, i.e. LC time constant of this node, $\tau_{LC} = \sqrt{LC}$, is insignificant. Hence our LC node reduction is basically a process to eliminate insignificant LC time constants.

Given a $RLKC$ line, its damping factor is defined by $\xi = \frac{Rl}{2}\sqrt{\frac{C}{L}}$, where R, $L$and $C$are the resistance, inductance and capacitance per unit length of the line, respectively, and $l$is the length of th line. Alternatively, damping factor can be expressed as $\xi = \frac{R_t C_t}{2\sqrt{L_t C_t}} = \frac{\tau_{RC}}{2\tau_{LC}}$, where R_t, L_t, and $C_t$are total resistance, inductance and capacitance of the line, respectively, and $\tau_{RC}$and $\tau_{LC}$are the RC and LC time constants of the line. Because applying our reduction algorithm will not change the total resistance, inductance and capacitance, hence damping factor and RC, LC time constant of the $RLKC$ line will be kept unchanged.

4. EXPERIMENTAL RESULTS

We implement all our reduction algorithms which is shown in Table 1 in C language and runs on a PIII $900MHz$ machine with $256MB$ memory.

The simulation results of different circuits are listed in Table 2. Figure 8 plots the waveforms of c397 before and after reduction. It also gives the error curve which shows that the maximum error is less than $50mV$.

Figure 9 and 10 show the detection of crosstalk and overshoot by applying our $RLKC$ reduction algorithm.

Figure 11 shows the runtime and memory consumption of our algorithm for different circuits. It shows that the runtime and memory consumption of our algorithm are roughly linear proportional to circuit size.

5. CONCLUSION

In this paper, we present an efficient realizable reduction algorithm for tightly coupled $RLKC$ interconnect circuits. Several effective reduction rules have been introduced. The experimental results show that the runtime of our algorithm is approach to near and achieved within 3% of errors and over 50% reduction ratio.

6. REFERENCES

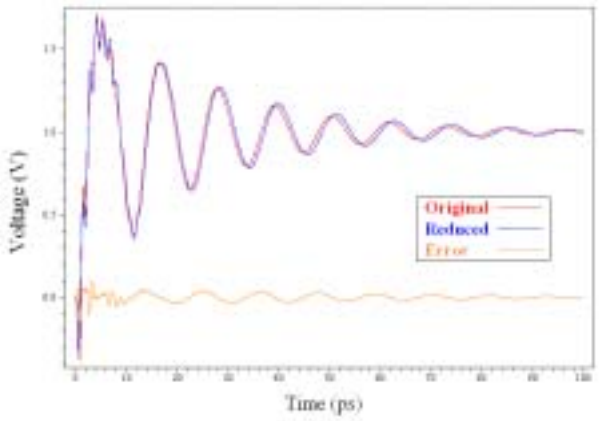

Figure 8: Waveforms Before/After Reduction and Error Curve.

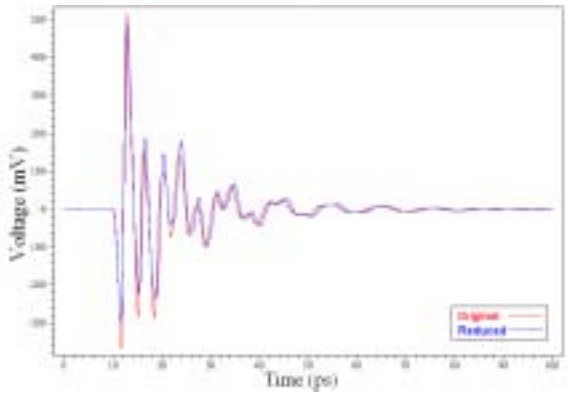

Figure 9: Detection of Crosstalk Voltage Glitch.

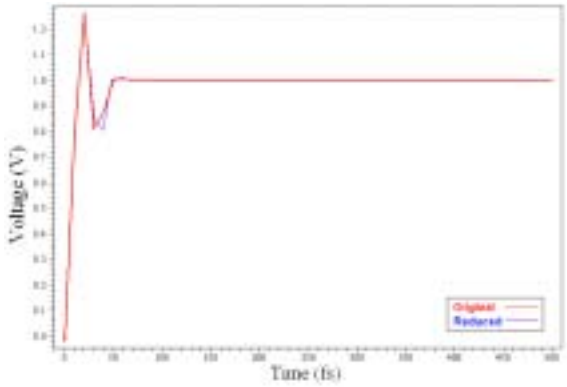

Figure 10: Detection of Overshoot.

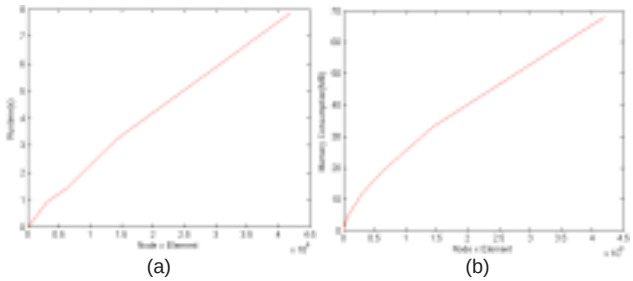

Figure 11: (a) Running Time; (b) Memory Consumption.

[1] A. Devgan and P. O'Brien. Realizable reduction for rc interconnect circuits. *Proc. ICCAD*, 1999.

[2] P. Feldmann and R. W. Freund. Efficient linear circuit analysis by pade via lanczos process. *IEEE Trans. Computer Aided Design*, pages 639–649, May 1995.

[3] K. J. Kerns, I. L. Wemple, and A. T. Yang. Stable and efficient reduction of substrate model networks using congruence transforms. *Proc. ICCAD*, pages 207–214, November 95.

[4] B. Krauter and S. Mehrotra. Layout based frequency dependent inductance and resistance extraction for on-chip interconnect timing analysis. *Design Automation Conference, 1998*, 1998.

[5] A. Odabasioglu and L. Pillage. Prime:passive reduced-order interconnect macromodeling algorithm. *Proc. ICCAD'97*, pages 58–65, November 1997.

[6] L. Pillage and R. Rohrer. Asymptotic waveform evaluation for timing analysis. *IEEE Trans. Computer Aided Design*, 9:352–366, April 1990.

[7] P. Rabiei and M. Pedram. Model order reduction of large circuits using balanced truncation. *DAC*, February 1999.

[8] B. Sheehan. Ticer:realizable reduction of extracted rc circuits. *Proc. ICCAD*, 1999.

[9] Q. Yu and E. S. Kuh. Passive time-domain model order reduction via orthonormal basis functions. *Proc. ECCTD'01*, September 2001.