RLC Reduction Scheme for Modeling Interconnection Line Delay in Nano-CMOS Circuits

Chanyong Jung, Sukin Yoon and Taeyoung Won

Department of Electrical Engineering and National Center for Computational Electronics, Inha University
253 Yonghyun-dong, Nam-gu, Incheon, Korea 402-751
Phone: +82-32-860-8686 Fax: +82-32-875-7436 E-mail: {jcy, siyoon, twon}@hsel.inha.ac.kr

ABSTRACT

In this paper we propose a realizable RLC-in-RLC-out technique to reduce parasitic parameters. The proposed technique is an efficient MOR (Model Order Reduction) method, which makes it possible to control the rise and delay time errors within the limit corresponding to the maximum frequency of operation. In addition, the equivalent circuit with reduced number of node elements derived from the proposed algorithm does conserve the passivity due to the use of zero-th or first order approximation. The proposed algorithm from node elimination is based on TICER (Time Constant Equilibration Reduction) approach. The reduction is achieved by eliminating the so-called quick nodes which have time constant less than the user-defined time constant. The nodal time constant is a maximum value between the RC time constant and LC time constant. For an exemplary circuit with 19,600 elements, the simulation revealed that we have maximum 0.6 % error rate at 97 % reduction rates.

Keywords: realizable, parasitic, reduction, errors

1 INTRODUCTION

As device size gets smaller and circuits also get larger, a circuit design is more difficult. Circuit analysis with a number of parasitic elements is important to verify efficiency of IC (integrated circuit) designs. MOR (Model Order Reduction) is an efficient method for the analysis. AWE using Fade's approximation is best known method, oldest, and basis for other MOR methods [1]. However, it suffers primary problems. The AWE method can be unstable for high order approximation due to using low order approximation. Also, algorithms for tools such as PRIMA and RICE based on AWE are complex to be implemented. A realizable RC-in-RC-out technique becomes a promising candidate to overcome their drawbacks. In TICER, Sheehan suggested the criterion of nodal time constants and demonstrated that provided the time constant by eliminating nodes produce little error which is small compared to the rise and fall times of the RC circuit. But these methods only work for RC circuits. Recently the DTT method was proposed to approximate transfer functions in tree-structured RLC circuits by directly transfer functions truncations. Since a truncated transfer function may not be a real function, however, the algorithm is not agreed with tools such like SPICE.

The reduced circuit using the proposed method provides faster time and lower memory, and the algorithm for reduction is easy to implement. Also, it guarantees stability [2-4]. The reduced equivalent circuit using this proposed method saves memory, guarantees passivity, provides a good error boundary.

2 THEORY

Consider a parasitic RLC network with k nodes. In the RLC network with no current sources, the voltage of the node is needed the following equation at s-domain:

$$Y(s)V(s)=0 \tag{1}$$

where $Y(s)$ is n by n admittance matrix and $V(s)$ is corresponding matrix. The node x is the eliminating node and k is the number of elements connected to the first neighboring node. Where $Y_x = \sum_{j=1}^{k} y_j$, equation (1) is given by the following equation:

$$Y_x V_x - y_1 V_1 - y_2 V_2 - \ldots - y_k V_k = 0 \tag{2}$$

The node x with its three neighbors is showed in Fig.1. After the node x is eliminated, the neighbors is changed.

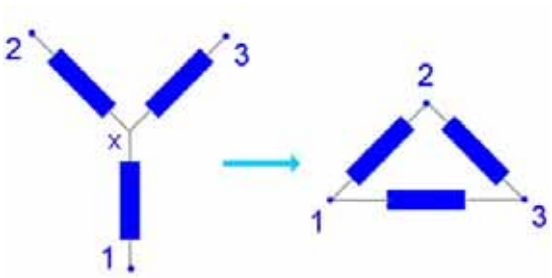

Fig. 1 Elimination of the node with three elements.

From the admittance network of the eliminated node x, equation (2) is used for elimination of Y_x . Then, V_x is

$V_x = \left[\sum_{j=1}^{k} y_j V_j \right] / Y_x$ and Equation of the first neighbor (node 1) of x is now given by:

$$\left(\hat{Y}_1 + \frac{\left(\sum_{j=2}^{k} y_1 y_j \right)}{Y_i} \right) V_1 - \frac{\left(\sum_{j=2}^{k} y_1 y_j V_j \right)}{Y_x} - \sum_{\substack{r=1 \\ r \neq i}}^{k_1} y_r V_r = 0 \quad (4)$$

where $\hat{Y}_1$ is the sum of all admittances from node 1 except to node x, k1 is the number of nodes connected to node 1.

Where $\hat{Y}_1 = \sum_{\substack{r=1 \\ r \neq i}}^{k1} y_r$, the above equation can be easily represented as followed:

$$[(\hat{Y}_1 + y_1(Y_i - y_1)/Y_i]V_1 - (\sum_{j=2}^{k} y_1 y_j V_j)/Y_i - \sum_{\substack{r=1 \\ r \neq i}}^{k1} y_r Y_r = 0 \quad (6)$$

After center node x of the network is eliminated, the new admittance induced by the neighbors of the node are $(\sum_{j=2}^{k} y_1 y_j)/Y_i$ and expressed as follows [2, 3]:

$$y_{pq} = y_p y_q / Y_x \quad (p, q = 1, 2, 3, \cdots, n) \quad (7)$$

where y_{pq} is the new admittance between node p and node q, Y_x is the sum of the all neighboring admittances incident to the node before the node x is not eliminated. In practical a circuit, one or more of R, L and C will be zero or infinite. The admittance connected to node x is ideally expressed as follows [2, 3]:

$$y_{pq} = \frac{y_p y_q}{Y_x} = \frac{\left(g_p + sc_p + \frac{b_p}{s} \right)\left(g_q + sc_q + \frac{b_q}{s} \right)}{\left(G_x + sC_x + \frac{B_x}{s} \right)} \quad (8)$$

where $G_x = \sum_{i=1}^{n} g_i$ is the sum of all conductances to the node x, $C_x = \sum_{i=1}^{n} c_i$ is the sum of all capacitances to the node x, $B_x = \sum_{i=1}^{n} b_i$ is the sum of all reciprocal inductances to the node x. In the circuit every node generally has three time constants (RC, LC, RL time constants). They are given by $\tau_{RC} = C_x / G_x$, $\tau_{LC} = \sqrt{C_x / B_x}$ and $\tau_{RL} = G_x / B_x$. For $sC_x / G_x > s\sqrt{C_x / B_x}$, the dominant time constant at the node x is the RC time constant. For $sC_x / G_x < s\sqrt{C_x / B_x}$, the dominant time constant at the node x is naturally the LC time constant. Note that we also consider RL time constant ($\tau_{RL} = G_x / B_x$) for paralleled RL branches although it is not used to eliminate nodes. The governing time constant at the node x can be expressed by $\tau_x = \max(\tau_{RC}, \tau_{LC})$ [2]. A user specified time constant is $\tau_{\min}$ and proportional to $1/(2\pi f_{\max})$. For $\tau_x < \tau_{\min}$, the node x becomes a quick node [3]. Equation (2) can be expanded into a polynomial expression in s-domain up to first order if the node x is a quick node. The constant coefficients of s parameters can be among a conductance, an inductance and a capacitance.

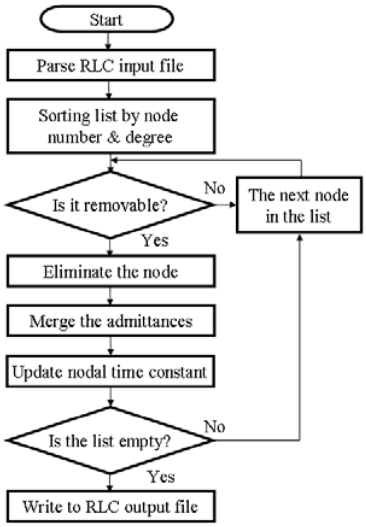

Fig. 1 The flowchart for RLC reduction algorithm.

The flowchart for RLC reduction algorithm is shown in Fig. 1.

3 SIMULATION RESULT

We implemented the reduction program in C++ programming language, reduced RLC circuits with 48 to 19,600 elements and two ports and simulated the original circuits and the reduced circuits using PSPICE.

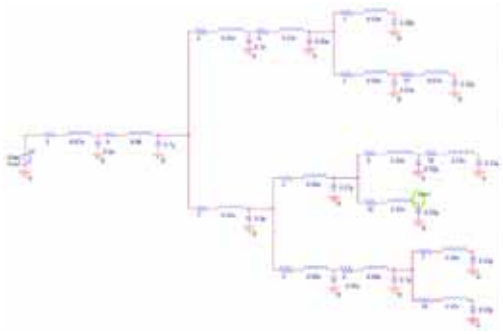

Fig. 2. The equivalent circuit for tree-like interconnect.

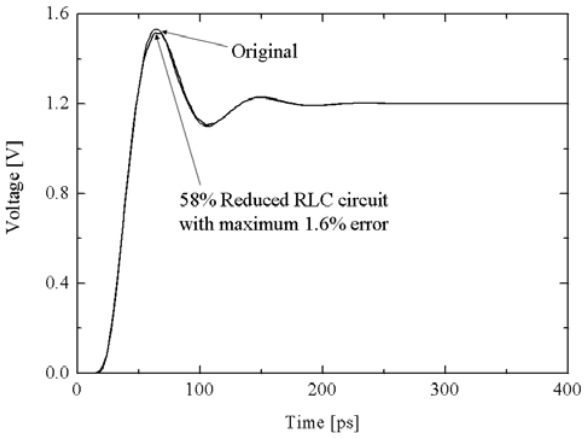

Fig. 3 Small RLC circuit with tree-like structure.

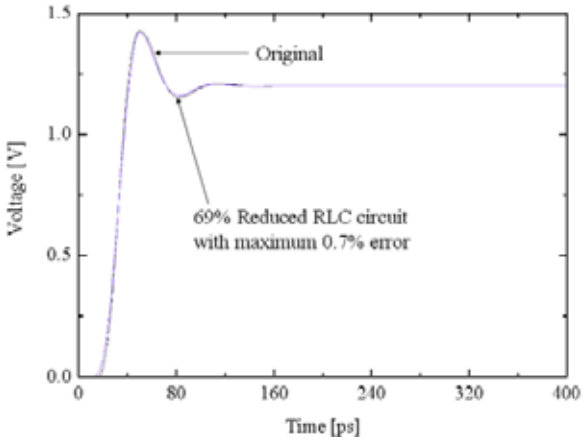

Fig. 4 Small RLC circuit with tree-like structure.

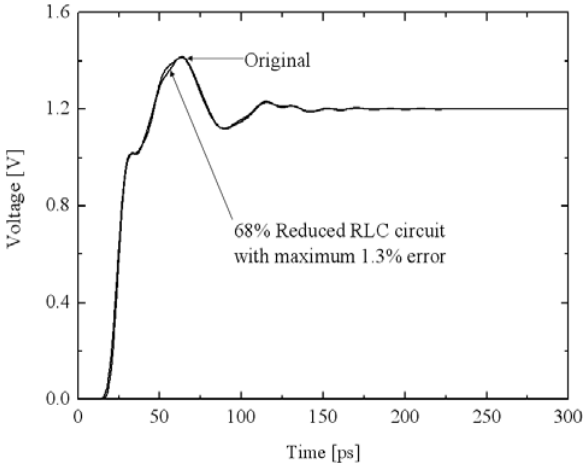

Fig. 5 Medium RLC circuit with mixed structure.

Fig. 2 shows schematic of original tree-like circuit consisted of resistors, capacitors and inductors. In Fig. 3 we plot the responses of the original tree-like circuit with 48 elements and the reduced circuit at 58% reduction rate with $\tau_{\min} = 1.5$ *ps*. The rise time errors were maximum 2.6 %. For the original circuit with 988 elements, reduction rates was 68 % with $\tau_{\min} = 0.9$ *ps*, the rise time error was 1.3 % (Fig. 5). The above results do provide a suitable $\tau_{\min}$ for small and medium sized circuits at resultant reduction rate. The transient response of the original mesh-like circuit with 82 elements and reduced circuit at 69% reduction rate is showed in Fig. 4. The rise time errors were maximum 0.7 % with $\tau_{\min} = 3.3$ *ps*.

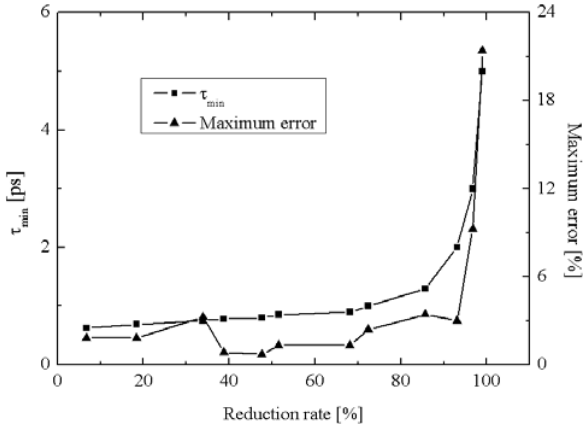

Fig. 6 $\tau_{\min}$ and maximum error rate vs. reduction rate.

The different levels of maximum error rate and $\tau_{\min}$ at reduction rate are shown in Fig. 6. It reveals that the higher

reduction needs the slower τ_{min} , induces at the higher error.

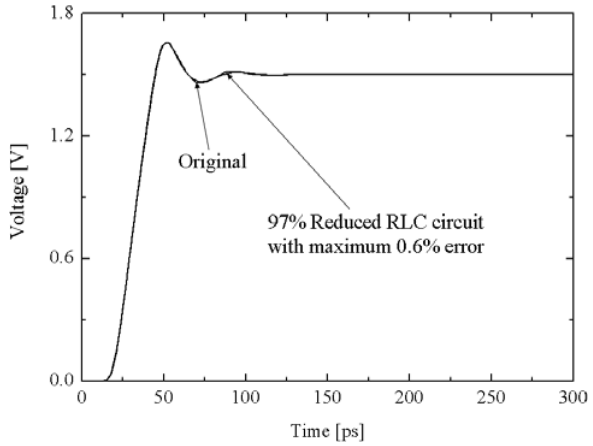

Fig. 7 Transient response from a large circuit with asymmetric tree-structure.

Fig. 7 shows the transient response of the large circuit with 19,600 elements and the circuit which was reduced down by 97 %. The output waveform for the reduced is almost identical to the waveform in the original circuit with 19,600 elements. There was nearly no loss of error in rise time. It suggests that the simulation using the proposed reduction algorithms a good result of a large circuits at a very high reduction rate.

Table 1. CPU runtime of reduced circuits as compared to original circuits for four structure circuit.

Circuit [#]		CPU time [s]	
Nodes	Elements	Original	Reduced
3400	4800	5.41	1.39
13872	19600	111.92	44.58
40800	57600	1086.02	554.19

Table 1 shows CPU runtime of the SPICE simulations with the reduction algorithm and without it, respectively. The reduced circuit was obtained at 99 % reduction rate.

4 CONCLUSION

This paper presented a reduction algorithm for a circuit including parasitic RLC network. It is realizable, efficient and simple. All standard simulators can handle the reduced circuits without any modification because the proposed realizable reduction method is RLC-in-RLC-out. Our simulation results show that the proposed method can achieve high reduction ratio with specified frequency and provide the valid accurate result with reduced CPU runtime for circuit designers.

REFERENCE

[1] L. Rohrer and L. Pillage, “Asymptotic Waveform Evaluation for Timing Analysis,” IEEE TCAD, vol. 9, pp 352-366, 1990.

[2] C. Amin, M. Chowdhury, Y. Ismail, “Realizable RLCK Circuit crunching,” DAC 2003, pp. 226-231, 2003.

[3] Bernard N. Sheehan, “TICER: Realizable Reduction of Extracted RC Circuits,” Digest of Technical Papers, IEEE/ACM Proc. of ICCAD, pp. 200-203, 1999.

[4] Z. Qin, C. K. Cheng, “Realizable parasitic reduction using generalized Y-delta transformation,” DAC 2003, pp 220-225, 2003.

[5] Bernard N. Sheehan, “Branch Merge Reduction of RLCM Networks,” IEEE/ACM Proc. of ICCAD, pp 658-664, 2003.