

Realizable Parasitic Reduction Using Generalized Y- Δ Transformation

Zhanhai Qin
Synopsys, Incorporation
zhanhai.qin@synopsys.com

Chung-Kuan Cheng
CSE Dept. Univ. of California, San Diego
kuan@cs.ucsd.edu

Abstract— We propose a realizable RCLK-in-RCLK-out parasitic reduction technique. The method employs generalized Y- Δ transformation. In our method, admittances are kept in their original rational forms of s , and their orders are reduced by truncating high-order terms. Therefore reduced admittances match the low-order terms in exact admittances. First-order realization of admittances is guaranteed, and higher-order realization is achieved by template optimization using Geometric Programming. The algorithm uniquely uses common-factor identification and cancelation operations to make Y- Δ transformation numerically stable. The experiment shows that our method can achieve higher reduction ratio than TICER and comparable simulation results with PRIMA.

Categories & Subject Descriptors: B.7.2 Simulation, B.8.2 Performance Analysis & Design Aids, G.2.2 Graph Theory.

General Terms: Algorithms, Performance, Design.

Keywords: Y- Δ Transformation, model order reduction, parasitic reduction.

1. INTRODUCTION

With ever increasing design complexity huge amount of extracted interconnect data size has pushed the capacity of existing timing/noise analysis and transistor level simulation tools to the limits. Recent work in the model order reduction has been focused on generating stable and passive macromodels [1]–[6]. However, these models cannot directly be fed into a general simulator. Integrating realizable reduction techniques into design flow of real applications shows more advantages. For example, RC-in-RC-out like reduction technique has been used widely in the extraction and transistor level simulation stage.

Liao [7] proposes a method to realize reduced RC macromodels from progressively merged sub-circuits. Sheehan [8] presents an RC-in-RC-out reduction scheme named TICER, in which internal nodes with extremely large and small time constants are eliminated using Gauss eliminations. A nice property of the methods is that reduced RC circuits can be plugged back into the system and simulated using any general simulator. At the same time they preserve the zero and first order of moments. But these methods only work for RC networks.

In another aspect, topological analysis [9] is an approach to calculating driving-point admittances using Cramer's rule in s -domain. The determinant of an admittance matrix of a passive network without mutual inductances is equal to the sum of all the tree admittance products of the network. Hence the method avoids cancelations in

This work was supported in part under grants from NSF project number MIP-9987678 and the California MICRO program.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DAC 2003, June 2-6, 2003, Anaheim, CA, USA.

Copyright 2003 ACM 1-58113-688-9/03/0001 ...\$5.00.

the expansion of determinants. However, enumerating all the trees in a large network is very difficult.

Recently Ismail proposes DTT to approximate transfer functions in tree-structured RLC networks by direct transfer-function truncations [10]. The transfer functions are kept in the rational form so that low-order moments are matched implicitly. Since a truncated transfer function may not be a positive real function, the method is not compatible with general simulators either.

Unsatisfied with these limitations, we devise a new RCLK-in-RCLK-out reduction method based on Y- Δ transformation [11] [12] for general RCLK-VJ¹ linear networks. The idea is that, given such a network, we perform Y- Δ transformation on all internal nodes in an efficient order. Y- Δ admittances in each transformation are kept in the rational form, and admittances whose order is higher than a threshold β will be truncated. Unlike topological analysis and other symbolic approaches, the new reduction method only evaluates the first β terms in the denominator and numerator of Δ admittances.² Discarded high-order terms, however, do not affect the preservation of the low-order terms, i.e., the low-order terms are precisely the first $\beta + 1$ terms in exact admittances.

The main contribution of this paper:

- 1) Realizability of reduced RCLK network is achieved via Y- Δ transformation and Geometric Optimization;
- 2) Fidelity of low-order terms in Δ admittances are kept, and the orders of Δ admittances are reduced to no more than β ;
- 3) First $\beta+1$ moments of exact admittances are matched implicitly by reduced Y- Δ admittances, including the zero order moment for DC correctness;
- 4) Two kinds of common-factor effects are first discovered in Y- Δ transformation. The findings lead to *essential* numerical improvement to Y- Δ transformation.

The remaining of the paper is organized as follows. Generalized Y- Δ transformation formula are given in Section 2. Common-factor effects are illustrated and solved in Section 3. In section 4, we present the RLC realization method using Geometric Programming. Section 5 reviews the proposed reduction algorithm. Section 6 shows examples and experimental results.

2. GENERALIZED Y- Δ TRANSFORMATION

First we clarify the assumptions and notations we will use throughout the paper. We confine our discussion to linear RCLK-J networks. We assume that all current sources are shunt to ground and all storage elements have no initial conditions.³

- n_k denotes the k -th labeled node, where k starts from 0. Nodes are eliminated in the order labeled;

¹A RLCK-VJ network contains resistors, capacitors, inductors, K elements, independent voltage and current sources only.

²Y and Δ admittances are referring to inputs and outputs of a Y- Δ transformation.

³Voltage sources and floating current sources can be converted to shunt current sources using source transformation, and initial conditions can be simply modeled as constant current or voltage sources in s -domain.

- when the k -th node is eliminated, the network will be updated accordingly. We label the network (graph) before the elimination as $G_k(V_k, E_k)$, and the network (graph) after as $G_{k+1}(V_{k+1}, E_{k+1})$;
- (n_i, n_j) denotes the branch between n_i and n_j , $Y_{i,j}$ the admittance of (n_i, n_j) , I_i the current source impinging on n_i , and Γ_i the neighbor set of n_i . A superscript (k) on these notations stands for “in G_k ”;
- The first neighbor of n_i is the node in Γ_i with the smallest label.

Our Y- Δ transformation is generalized in the following three ways:

- 1) it identifies and cancels common factors present in Δ admittances to assure numerical stability;
- 2) it covers circuits of general topology in contrast to DTT, which is tree-based. Aside from RLC, it also covers mutual K elements and current sources, which call for more sophisticated source transformation because of common-factor effects.
- 3) it eliminates nodes efficiently by dynamically choosing the one with the minimum degree. In TICER, however, the degree of quick and slow nodes is not a concern in its ordering scheme.

We cover K-element transformation in the next sub-section, followed by Y- Δ transformation formula. Common-factor effects will be covered in the next section.

2.1. Conversion of Mutual K Branch

RLC can be handled directly by Y- Δ transformation because they have well-known admittance forms in s -domain. Mutual inductors, on the contrary, have no simple admittance form. Because the branch voltage v_1 of a mutual inductor relies on current variations in more than branches including itself

$$V_1 = L_{11}I_1s + L_{12}I_2s + \dots$$

Including mutual inductors is difficult, because we are not able to eliminate a node if one of its incident branch inductively couples with more than one branch.

Alternatively, with K elements, inductive coupling can be modeled in such a way that variation of a branch current replies on multiple coupling branch voltages. I - V relationship of mutual K elements is given by

$$\frac{K_{11}}{s}V_1 + \frac{K_{12}}{s}V_2 + \dots = I_1, \quad (1)$$

where V_i 's are branch voltages.

In order for us to invoke Y- Δ transformation on arbitrary nodes, we still have to change mutual-K elements to self-K elements. In

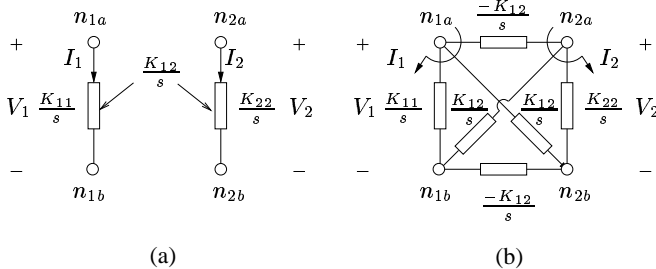


Fig. 1. Conversion on mutual K in s -domain: (a)given mutual K; (b) converted self K.

Fig. 1(a), the KCL equations for the four nodes in terms of V_1 and V_2 can be written as

$$\begin{aligned} \frac{K_{11}}{s}V_1 + \frac{K_{12}}{s}V_2 &= I_1 \\ \frac{K_{12}}{s}V_1 + \frac{K_{22}}{s}V_2 &= I_2. \end{aligned} \quad (2)$$

One can check that the KCL equations for the four nodes in Fig. 1(b) are exactly the same as (2), so that (b) is equivalent to (a). However, (b) has only self-K elements. Although some values in (b) are negative, the circuit is still passive because K-based method guarantees the extracted K matrix to be positive definite.

2.2. Generalized Formula for Y- Δ Transformation

The generalized Y- Δ transformation formula for linear RCLK-VJ networks are given below.

Theorem 1 (Y- Δ Formula): Suppose n_k in G_k is the node being eliminated. $\forall n_i, n_j \in n_k$'s neighbors $\Gamma_k^{(k)}$, if branch (n_i, n_j) is not in G_k , then it will be added into G_{k+1} after n_k is eliminated; otherwise a new admittance will be added on to the branch in G_{k+1} . Admittance $Y_{ij}^{(k+1)}$ for (n_i, n_j) is calculated as

$$Y_{ij}^{(k+1)}(s) = Y_{ij}^{(k)}(s) + \frac{Y_{ik}^{(k)}(s) \times Y_{jk}^{(k)}(s)}{\sum_l Y_{lk}^{(k)}(s)}, \quad \forall n_l \in \Gamma_k^{(k)}. \quad (3)$$

If $I_k^{(k)} \neq 0$,

$$I_i^{(k+1)}(s) = I_i^{(k)}(s) + \frac{Y_{ik}^{(k)}(s)}{\sum_l Y_{lk}^{(k)}(s)} I_k^{(k)}(s), \quad \forall n_l, n_i \in \Gamma_k^{(k)}. \quad (4)$$

For admittances and current sources not mentioned above, they will be inherited from G_k to G_{k+1} .

The theorem states that when we perform Y- Δ transformation on n_k , neighbors of n_k in G_k will become pairwise adjacent in G_{k+1} . And if n_k in G_k has a current source, then each of its neighbors will have a current source in G_{k+1} . It can be proven by solving for v_k from the KCL of n_k in terms of other voltage variables and substituting the solution for v_k in other KCL equations. The proposed transformation is different from Gauss elimination, because we calculate $Y_{ij}^{(k+1)}$ in (3) only up to the term of order β . Since computation of higher-order terms is skipped, we get an approximation of $Y_{ij}^{(k+1)}$ whose numerator and denominator are equal to the first β terms in $Y_{ij}^{(k+1)}$'s numerator and denominator, respectively. Stable admittances and transfer functions can be obtained from Δ admittances because of this nice property [14]. This further treatment is not in DTT.

Corollary 1: If all RLC elements in a linear network are positive, $Y_{ij}^{(k+1)}$ in (3) is a rational function of s

$$Y_{ij}^{(k+1)}(s) = \frac{A_{ij}^{(k+1)}}{B_{ij}^{(k+1)}} = \frac{\sum_{p=0}^m a_p s^p}{\sum_{q=0}^n b_q s^q}, \quad (5)$$

and a_p, b_q in (5) are non-negative.

Corollary. 1 holds because no subtraction is introduced in (3). Note that all the coefficients of powers of s in (5) are computed numerically.

3. COMMON-FACTOR EFFECTS

Common factors are introduced into the numerator and denominator of new admittance and current source in (3) and (4). They are harmful because 1) they cause exponential growth of the magnitude of coefficients in the numerators and denominators; 2) they create fake zeros/poles that hamper pole/zero approximation. Unfortunately, common-factor effects are present even though we perform truncations. First we go through an example to show when these common factors are generated and what they are composed of. Then we will give theorems for their existence. The impact of common-factor effects on Y- Δ admittances is discussed in the experiments.

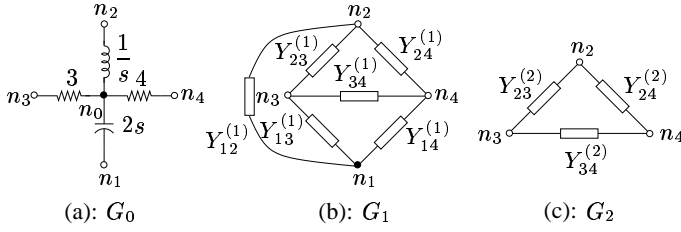


Fig. 2. A numerical example showing common factor existence: (a) n_0 is to be eliminated; (b) n_1 is to be eliminated; (c) n_0 and n_1 eliminated.

3.1. Example on Common-Factor Effects

A second-order circuit is given in Fig. 2(a). We want to apply Y- Δ transformation on n_0 and n_1 orderly. First we eliminate n_0 using (3) in Fig. 2(a). After the transformation, we have six admittances in (b):

$$\begin{aligned} Y_{12}^{(1)} &= \frac{2s}{\omega} & Y_{14}^{(1)} &= \frac{8s^2}{\omega} & Y_{24}^{(1)} &= \frac{4}{\omega} \\ Y_{13}^{(1)} &= \frac{6s^2}{\omega} & Y_{23}^{(1)} &= \frac{3}{\omega} & Y_{34}^{(1)} &= \frac{12s}{\omega}, \end{aligned} \quad (6)$$

where $\omega \equiv 1 + 7s + 2s^2$. Note that ω is the common denominator of the new admittances in (6).

Next we eliminate n_1 in (b). The admittance in (c) is computed again using (3). for instance,

$$Y_{23}^{(2)} = Y_{23}^{(1)} + \frac{Y_{12}^{(1)} \times Y_{13}^{(1)}}{Y_{12}^{(1)} + Y_{13}^{(1)} + Y_{14}^{(1)}}. \quad (7)$$

Insert (6) into (7),

$$Y_{23}^{(2)} = \frac{6s + 42s^2 + 12s^3}{\omega(2s + 14s^2)}. \quad (8)$$

Note that we have considered ω as a common denominator, so that the result in (8) is simplified. Without identifying this common denominator, we would have

$$Y_{23}^{(2)} = \frac{6s\omega^2 + 18s^2\omega^2 + 24s^3\omega^2 + 12s^3\omega^2}{\omega(2s\omega^2 + 6s^2\omega^2 + 8s^2\omega^2)}. \quad (9)$$

ω in (6) is called ‘‘type-I common factor’’, when it is shared explicitly among denominators of admittances incident to the node being eliminated, e.g., n_1 in Fig. 2(b). Note that the existence of type-I common factor is not straightforward in the case that n_1 is also adjacent to nodes besides n_0 ’s neighbors when n_1 is eliminated, and in the case that other nodes are eliminated before n_1 and after n_0 , when admittance merging happens. In the two cases, ω is generally just a *factor* shared by the denominators of *some* of the admittances incident to n_1 .

To show the other kind of form of common factor ω , we go back to look at (8). Interestingly, ω is present in not only the denominator, but also the numerator, because $6s + 42s^2 + 12s^3$ can be factorized as $6s(1 + 7s + 2s^2) = 6s\omega$. Therefore, ω is common factor shared by the denominator and numerator of $Y_{23}^{(2)}$. We compute $Y_{24}^{(2)}$ and $Y_{34}^{(2)}$ in the same way, and it turns out that ω is present in their numerators and denominators as well. Because the presence of ω is implicit in the numerators, we name it as ‘‘type-II common factor’’.

In summary, we have found out that *Type-I common factor emerges explicitly in the denominator of every new admittance immediately after n_0 is eliminated. Type-II postpones until n_1 — the first neighbor of n_0 is eliminated, when type-II emerges implicitly in both the numerator and denominator of every new admittance induced by n_1 . Both types of common factors need to be treated for numerical stability concerns.* Sub-section 3.2 to 3.5 clarify the composition of common factors, and their existence in Δ admittances and current

sources for general linear network reduction using the proposed Y- Δ transformation. We will make the theorems easy to follow. Rigorous proofs can be found in [13].

3.2. Composition of Common Factor

Theorem 2: Suppose n_k is to be eliminated, n_k has m neighbors in G_k , and the admittances of branches incident to n_k are denoted as $\frac{A_1}{B_1}, \frac{A_2}{B_2}, \dots, \frac{A_m}{B_m}$. Type-I and type-II common factors associated with n_k are equal to ω_k , which is defined as:

$$\omega_k = \frac{\sum_{i=1}^m \left(A_i \prod_{j=1, j \neq i}^m B_j \right)}{W_k}, \quad \text{where } W_k = \prod_{i=0}^{k-1} \omega_i^{p_i}. \quad (10)$$

p_i is the number of denominators in $\{B_1, B_2, \dots, B_m\}$ with factor ω_i .

Th. 2 defines the composition of type-I and type-II common factors⁴ in a recursive way, based on Th. 1. It is derived from the denominator of the new admittance⁵ in (3), and we have simplified it by considering type-I common factors in denominators B_1, \dots, B_m . Each node has a unique ω which emerges at different times in different forms: it appears explicitly first in denominators as a type-I common factor, and it appears implicitly later in numerators and explicitly in denominators as a type-II common factor. ω ’s are atomic common factors. W_k defined in (10) are composed of one or more different atomic common factors.

3.3. Existence of Type-I Common Factor

A Type-I common factor associated with any node n_k appears immediately after it is eliminated. We figure out that ω_k in (10) defines a factor that the denominator of every new admittance will have after n_k is eliminated. Even when admittance merging happens, i.e., $Y_{ij}^{(k)} \neq 0$, the factor is still with the denominator of the resultant admittance $Y_{ij}^{(k+1)}$. Because when two admittances are merged, their denominators are multiplied together. So that a factor in any denominator before the merging is still a factor in the new denominator after it. We formally state the observation in Th. 3.

Theorem 3 (Existence of Type-I Common Factor): After n_k in G_k is eliminated, $\forall n_i, n_j \in n_k$ ’s neighbors $\Gamma_k^{(k)}$, the denominator of new admittance $Y_{ij}^{(k+1)}$ in G_{k+1} will have a factor ω_k associated with n_k .

3.4. Existence of Type-II Common Factor

A Type-II common factor associated with any node n_k appears in both numerators and denominators of new admittances after n_k ’s first neighbor is eliminated. Identifying the existence of type-II common factors is very difficult, because they appear in numerators implicitly. The proof of their existence is a key step towards an effective and efficient Y- Δ transformation for our reduction purpose.

Theorem 4 (Existence of Type-II Common Factor): Suppose n_f is the first neighbor of n_k in G_k . After n_k and n_f are eliminated, $\forall n_i, n_j \neq n_f \in n_k$ ’s neighbor set, the numerator and denominator of new admittance $Y_{ij}^{(f+1)}$ in G_{f+1} has a common factor ω_k associated with n_k .

Theorem 5 (Recursive Existence of Type-II Common Factor): Suppose n_f is the first neighbor of n_k in G_k , and ω_k is associated with n_k . Th. 4 holds no matter how many nodes are eliminated and how many type-II common factors are cancelled in Δ admittances between n_k ’s and n_f ’s eliminations.

Similar to mathematical induction, Th. 4 lays the foundation of the existence of type-II common factors, and Th. 5 affirms that it is safe to cancel type-II common factors at any time after they appear.

⁴Common factor associated with a node changes according to different β (order of truncation). But the common-factor effects are persistent, $\forall \beta > 0$.

⁵It refers to the admittance on the right in (3) before the merging.

3.5. Existence of Common Factor in Current Source Transformation

Current source transformation is performed along with Y- Δ transformation. Common-factor effects in the latter occur in the former as well. We summarize them in Th. 6.

Theorem 6: Suppose n_f is the first neighbor of n_k in G_k , and ω_k is associated with n_k .

- 1) If n_k has a current source $I_k^{(k)}$, then after n_k is eliminated, $\forall n_i \in n_k$'s neighbors $\Gamma_k^{(k)}$, the denominator of new current source $I_i^{(k+1)}$ in G_{k+1} will have a factor ω_k ;
- 2) After n_k and n_f are eliminated, $\forall n_i \in n_k$'s neighbors $\Gamma_k^{(k)}$ in G_{f+1} , the numerator and denominator of new current source $I_i^{(f+1)}$ has a common factor ω_k . The statement holds no matter how many nodes are eliminated and how many type-II common factors are cancelled in Y- Δ admittances and transformed current sources between n_k 's and n_f 's eliminations.

4. Δ ADMITTANCE REALIZATION

After eliminating all internal nodes using Y- Δ transformation, we realize each Δ branch by calling for a positive real function approximation. Since all the coefficients in Δ admittances are non-negative, first order realization is guaranteed. Therefore Elmore delays of original networks are preserved. Higher order realization of Δ admittances⁶ are achieved by choosing a passive template structure with the same order and do the approximation with the constraints that each element needs to be positive and the moments are kept approximately the same.

4.1. First Order Realization

For any second-order Δ admittance of the form

$$Y_{ij}(s) = \frac{a_0 + a_1 s}{b_0 + b_1 s}, \quad (11)$$

it is seen from Corollary 1 that

$$a_0, a_1, b_0, b_1 \geq 0.$$

If $a_0 b_1 \leq a_1 b_0$, $Y_{ij}(s)$ in (11) can be realized using circuit in Fig. 3(a), where

$$R_1 = \frac{b_1}{a_1} \quad R_2 = \frac{a_1 b_0 - a_0 b_1}{a_0 a_1} \quad C = \frac{a_1^2}{a_1 b_0 - a_0 b_1}; \quad (12)$$

otherwise it can be realized using circuit in (b), where

$$R_1 = \frac{b_0}{a_0} \quad R_2 = \frac{a_0 b_1 - a_1 b_0}{a_0 a_1} \quad L = \frac{a_0 b_1 - a_1 b_0}{a_0^2}. \quad (13)$$

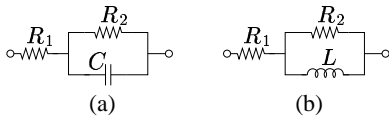


Fig. 3. First Order Realization: (a) RC configuration; (b) RL configuration

Consider RC networks with each branch consisting of a conductance g_i and capacitance c_i in parallel. Some elements may be absent, in which case the corresponding g_i or c_i is zero. For such a network with two branches, the Δ admittance after the center node is eliminated is

$$Y_{12}(s) = \frac{g_1 g_2 + (g_1 c_2 + g_2 c_1) s}{(g_1 + g_2) + (c_1 + c_2) s}.$$

It can be seen that

$$g_1 g_2 (c_1 + c_2) \leq (g_1 c_2 + g_2 c_1) (g_1 + g_2),$$

⁶ Δ admittances can be changed to stable admittances before the realization is called.

so that Y_{12} can be realized using RC model shown in Fig. 3(a). It can be proven by induction that any Δ admittance reduced from any RC network can be realized by the RC model.

4.2. Template Realization

To achieve high-order approximation to Δ admittances, we devise high-order templates borrowed from circuit structures in admittance realization theory. For example, the circuit shown in Fig. 4 is a structure used in Brune's synthesis procedure [15]. When serial-parallel withdrawal does not work for a realizable admittance, one can always invoke Brune's procedure to realize it using a similar structure. The method, however, can not be applied directly to Δ admittances, because these admittances are truncated so that they may not be realizable. But since the Δ admittances preserve low-order terms of exact input admittances, which are positive real functions, we can expect that the Δ admittances are positive and real within some moderate frequency range. This inspires us to use the general Brune's admittance structure as templates to approximate the Δ admittances, with the constraint that each element needs to be positive and the objective that the coefficients of the Δ admittances are matched as much as possible.

In Fig. 4, the shunt RLC box repeats the same circuit topology from the left, but elements may have different values. It is there because Brune's realization process may have multiple cycles. In each cycle, it uses the network on the left to the RLC box to reduce the order of the input admittance $Y(s)$ by two, until the RLC box can be realized by a conductance. In Brune's realized networks, L_1 is allowed to be either positive or negative, and L_3 always has an opposite sign to L_1 . All other elements in Fig. 4 are positive. Admittance in the form of Fig. 4 is a positive real function of s , and the T inductor series can be replaced by an ideal transformer with positive primary and secondary inductances.

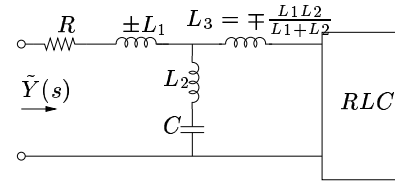


Fig. 4. Brune's Admittance Structure

We formulate the realization problem in Geometric Programming [16]. Suppose $Y(s)$ is a Δ admittance in the form of

$$Y(s) = \frac{\sum_0^\beta a_i s^i}{\sum_0^\beta b_i s^i}.$$

Here a_i and b_i are positive real numbers. We build a β -th order Brune's admittance $\tilde{Y}(s)$ in Fig. 4, $\tilde{Y}(s)$ can be written as

$$\tilde{Y}(s) = \frac{\sum_{i=0}^{\beta} \tilde{a}_i s^i}{\sum_{i=0}^{\beta} \tilde{b}_i s^i}, \quad (14)$$

\tilde{a}_i and \tilde{b}_i are given by

$$\tilde{a}_i = \sum_{j=1}^{x(\tilde{a}_i)} p_{ij}, \quad \text{and} \quad \tilde{b}_i = \sum_{j=1}^{x(\tilde{b}_i)} q_{ij}, \quad (15)$$

where

$$p_{ij} = e_{ij} t_1^{\alpha_{ij1}} t_2^{\alpha_{ij2}} \dots t_\nu^{\alpha_{ij\nu}} \quad e_{ij} > 0, \quad (16)$$

and

$$q_{ij} = f_{ij} t_1^{\beta_{ij1}} t_2^{\beta_{ij2}} \dots t_\nu^{\beta_{ij\nu}} \quad f_{ij} > 0. \quad (17)$$

In (16) and (17), t_1, t_2, \dots, t_ν are element variables in the template $\tilde{Y}(s)$, $x(\tilde{a}_i)$ and $x(\tilde{b}_i)$ are the numbers of p_{ij} and q_{ij} in \tilde{b}_i and

\tilde{a}_i , respectively. Note that they are not variables. Once a template is determined, $x(\tilde{a}_i)$ and $x(\tilde{b}_i)$ are known. We formulate the Geometric Programming problem as follows:

Objective function:

$$\min \left(\sum_{i=0}^{\beta} \sum_{j=0}^{x(\tilde{b}_j)} \frac{1}{p_{ij}} + \sum_{i=0}^{\beta} \sum_{j=0}^{x(\tilde{a}_j)} \frac{1}{q_{ij}} \right) \quad (18)$$

subject to

$$t_j > 0, \quad j = 1, 2, \dots, \nu \quad (19)$$

$$\tilde{a}_i \leq a_i, \quad \tilde{b}_i \leq b_i \quad i = 0, 2, \dots, \beta. \quad (20)$$

For example, given a 2nd-order Δ admittance

$$Y(s) = \frac{a_0 + a_1 s + a_2 s^2}{b_0 + b_1 s + b_2 s^2}, \quad (21)$$

we substitute R_1 for R and R_2 for the RLC box in Fig. 4 and use the resultant network as a template. Hence the input admittance $Y(s)$ of the template is

$$\tilde{Y}(s) = \frac{LR + (L_1^2 + CLR_1R_2)s + (CL_2^2R_1 + CL^2R_2)s^2}{L + CLR_2s + CL_2^2s^2}, \quad (22)$$

where $R \equiv R_1 + R_2$ and $L \equiv L_1 + L_2$. In this case, p_{ij} and q_{ij} in (18) are product terms in the numerator and denominator of (22) respectively; t_j in (19) is elements in the template; \tilde{a}_i and \tilde{b}_i are symbolic coefficients of powers of s in (22), and a_i and b_i are numerical coefficients in (21). The objective function tries to minimize the reciprocals of p_{ij} and q_{ij} . On the other hand under the constraints that these terms can not be greater than the numerical coefficients in (21). When the objective function is minimized and the inequalities in the constraints all become equalities, then the coefficients in (21) are matched, and so are the moments.

In summary, first-order realization of Δ admittances is guaranteed, and high-order realization is accomplished by template optimization. The new realization procedure works effectively for templates of order ten or less in our experiments. And the merit of the method is that templates are realized and preserve electrical properties at the port simultaneously.

5. REDUCTION FLOW

PROCEDURE **reduction_flow**(*ckt*)

- 1 Transform voltage sources to current sources;
- 2 Decouple any floating current source in *ckt*;
- 3 WHILE (\exists internal node in *ckt*) DO
- 4 Pick node n_k with the minimum degree;
- 5 IF $I_k \neq 0$ THEN
- 6 CALL **current_source_transformation** for I_k ;
- 7 CALL **Y- Δ transformation** to eliminate n_k ;
- 8 Remove common factors from new admittances;
- 9 Remove n_k and its incident branches from *ckt*;
- 10 Realize all Δ admittances.

The proposed Y- Δ transformation is more efficient than LU decomposition in SPICE in two ways. First, it invokes LU decomposition only once in s -domain. But SPICE does it repeatedly due to varying time steps for stiff circuits. Secondly, the algorithm allows dynamical memory de-allocation, as branches of nodes eliminated are no longer needed and can be freed. As a result, the memory requirement grows up in the middle of the reduction process and goes down until it terminates. Memory requirement in SPICE grows monotonically until LU decomposition is completed. This also means less non-zero fill-ins, which result in faster decomposition.

6. EXAMPLES

We use a few examples in order to show the superiority of proposed realizable parasitic reduction in terms of orders of reduced models and reduction efficiency. The first example is a high-performance clock distribution circuit in a real design case. The circuit with 78564 nodes is a mixture of RC trees and meshes. Our method can reduce it to a simple RLC circuit with four nodes only. Circuit given in Fig. 5(b) is such a realization. In this case, if we use TICER to achieve

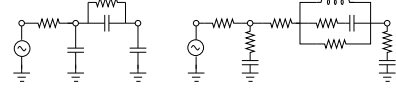


Fig. 5. Comparison of two reduced circuits

the same reduction ratio⁷, the reduced circuit is given in Fig. 5(a). Both methods realize each admittance using passive elements. But none of them in (a) matches first-order moments. In (b) the two shunt admittances to ground match to the first order, and the floating one matches to the second order. In Fig. 6, we use a reduced circuit with higher order branch admittance to generate the waveform. The waveform for TICER-reduced circuit (Fig. 5(a)) is also given in the figure for comparison. We find out that the waveform of Y- Δ are closer to the SPICE result without reduction.

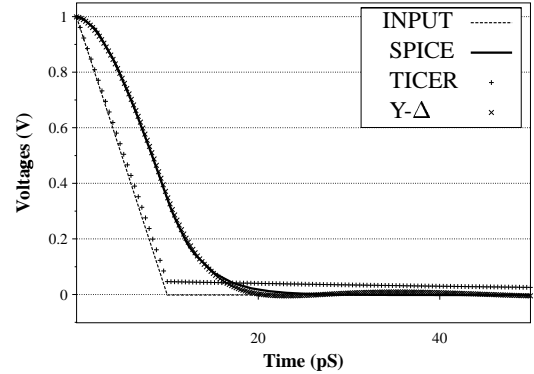


Fig. 6. Comparison of responses of a RC network

The response of a 62-node RLC network with 7 ports is plotted in Fig. 7. The response evaluated from the reduced network is very close to PRIMA's curve, and both of them are very close to SPICE output. Table I summarize the reduction ratio versus reduction efficiency. It

reduction ratio	Reduction Time(sec)	
	TICER	Y- Δ
20%	5.96	6.58
40%	23.58	23.79
60%	56.37	25.42
80%	249.33	27.76
99%	1634.78	46.56

TABLE I
EFFICIENCY COMPARISON

is clear that minimizing non-zero fill-ins is a good ordering scheme in Y- Δ transformation-based reduction techniques. For instance, TICER eliminates nodes by picking those with extreme time constants first. Because it does not consider the degrees of nodes when eliminating them, this ordering scheme is inferior in terms of efficiency when

⁷The original TICER algorithm uses time constant of each node as an error control mechanism, therefore such a high reduction ratio may not be allowed in the algorithm. The point of our comparison is that Y- Δ achieves a better reduction ratio, yet the result is close enough to SPICE.

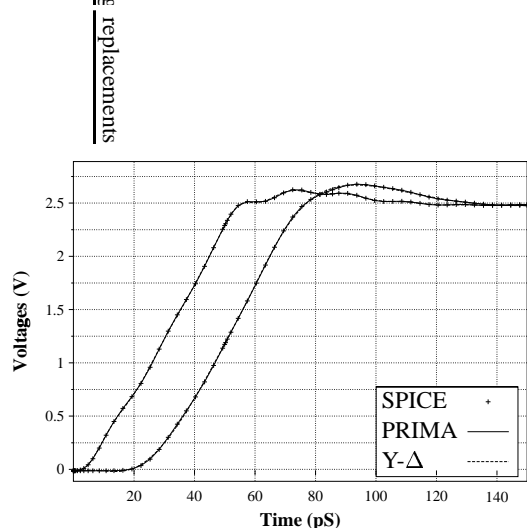


Fig. 7. Comparison of responses of a RLC network

high reduction ratio is preferred. Although one can specify a limit on reduction ratio in TICER, the complexity may be passed to the following simulation stage because the sparsity of circuits has been tampered already. A compromise of the two ordering schemes is also promising.

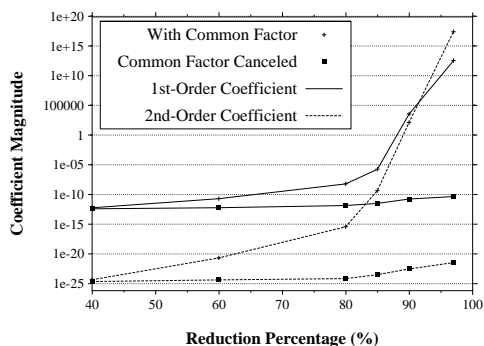


Fig. 8. Illustration of Common-Factor Effects

Fig. 8 compares the magnitude growth of coefficients in one branch admittance. The two ends of the branch are ports so that other admittances were merged onto it. The figure plots the coefficients collected from different powers of s . In the case when common factors are not canceled, they grow wildly along with reduction percentage, so that most of the actual significant bits are lost due to finite-precision computation. In fact, the higher the order is, the faster coefficients grow. For instance, the figure shows that the 2nd-order coefficient is smaller than the 1st-order one at low reduction ratio, but becomes larger at higher ratio. Because coefficients grow at different rate, the problem can not be solved by scaling. It is shown by the line with square dot that Y- Δ becomes practically useful only after common-factor cancelations are invoked.

As a special case of general Y- Δ transformation, DTT method does not raise common-factor question because they are tree-based. If nodes in a tree are eliminated in a bottom-up fashion, no non-zero fill-ins will be introduced. Therefore common factors do not exist in reduction of tree-structured circuits.

Fig. 9 plots locations of poles on complex plane. These poles are approximated for a power-ground design extracted at board-level with RLC elements. We leave only one current source (port) in the circuit because PRIMA's accuracy heavily depends on the number of ports. We find out that PRIMA does not return any complex poles. And although we have observed voltage oscillation in a wide frequency range, unfortunately SPICE is not able to do pole analysis on this stiff circuit either. However some low frequency poles are evaluated using Y- Δ transformation. The example shows the contribution of common-

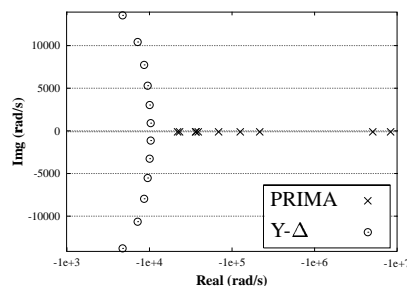


Fig. 9. Pole approximation

factor cancelations to numerical stability of Y- Δ transformation from another aspect.

7. CONCLUSIONS AND FUTURE WORK

We have proposed a realizable parasitic RCLK reduction method based on generalized Y- Δ transformation technique. Since the method is RCLK-in-RCLK-out, it is compatible with general simulators such as SPICE. The algorithm employs common-factor identification and cancelation operations to make Y- Δ transformation numerically stable. First-order realization of Δ admittances is guaranteed, and high-order realization is achieved by template optimization using Geometric Programming.

The algorithm can be easily extended to multi-port reduction. One can either stop Y- Δ reduction at any point and realize branches, or call for Δ -Y transformation, which is equivalent to LU's back solving. Like the forward Y- Δ transformation, the backward Δ -Y also has the two types of common factors to cancel out.

ACKNOWLEDGMENTS

The authors would like to thank Prof. E.S. Kuh for his advise on this work. They would also thank all the reviewers for their constructive comments and careful proofreadings.

REFERENCES

- [1] S. Lin, and E. S. Kuh, "transient simulation of lossy interconnects based on the recursive convolution formulation," *IEEE Trans. on CAS I: Fundamental Theory and Applications*, vol.39, pp.879-92, Nov. 1992.
- [2] P. Feldmann, R. W. Freund, "reduced-order modeling of large linear subcircuits via a block Lanczos algorithm," *Proc. of 32th DAC*, pp.376-80, 1995.
- [3] K. J. Kerns, I. L. Wemple, A. T. Yang, "stable and efficient reduction of substrate model networks using congruence transforms," *Proc. of ICCAD*, pp. 207-14, 1995.
- [4] A. Odabasioglu, M. Celik, L. T. Pillage, "PRIMA: passive reduced-order interconnect macromodeling algorithm," *IEEE Trans. on CAD*, vol.17, pp.645-54, Aug. 1998.
- [5] J. R. Phillips, L. Daniel, M. Silveira, "guaranteed passive balancing transformations for model order reduction," *Proc. of 39th DAC*, pp.52-7, 2002.
- [6] V. Raghavan, J. E. Bracken, R. A. Rohrer, "AWESpice: A general tool for the accurate and efficient simulation of interconnect problems," *Proc. of 29th DAC*, pp.87-92, 1992.
- [7] H. Liao, W. Dai, "partitioning and reduction of RC interconnect networks based on scattering parameter macromodels," *Proc. of 32th DAC*, pp.704-9, 1995.
- [8] B. N. Sheehan, "TICER: realizable reduction of extracted RC circuits," *Proc. of ICCAD*, pp.200-3, 1999.
- [9] S. Chan, "introductory topological analysis of electrical networks," *Holt, Rinehart and Winston, Inc.*, 1974.
- [10] Y. I. Ismail, E. G. Friedman, "DTT: direct truncation of the transfer function — an alternative to moment matching for tree structured interconnect," *IEEE Trans. on CAD*, vol. 21, No. 2, Feb. 2002.
- [11] S. B. Akers, Jr., "the use of wye-delta transformations in network simplification," *Operations Research*, 8, pp.311-23, 1960.
- [12] A.J. van Genderen, N. P. van der Meijis, "Using articulation nodes to improve the efficiency of finite-element based resistance extraction," *Proc. of 33rd DAC*, pp.244-51, 1996.
- [13] Z. Qin, C.K. Cheng, "linear network reduction via generalized Y- Δ transformation: theory," *Technical Report*, UCSD, CS2002-0706, 2002.
- [14] Z. Qin, C.K. Cheng, "RCLK-VJ network reduction with hurwitz polynomial approximation," *Proc. of Asian & South Pacific DAC*, pp. 283-91, 2003.
- [15] W. C. Yengst, "procedures of modern network synthesis," *Macmillan*, New York, 1964.
- [16] R. J. Duffin, E. L. Peterson, C. Zener, "geometric programming: theory and application," *John Wiley and Sons*, New York, 1967.