Reordering in Topology Decision Diagram Method for Symbolic Circuit Analysis

S. Djordjević and P. M. Petković

Abstract –This paper introduces reordering method in Topology Decision Diagram (TDD) in order to enhance symbolic analysis method for $RLCg_m$ network function generation in nested form. The improvement is obtained in circuit function compression and the execution time. An example of *n*-th order ladder network illustrates the method.

I. INTRODUCTION

Requirements for memory and CPU time consumption put the major limitations in symbolic analysis of the large circuits.

The technique of hierarchical decomposition results with compact symbolic expressions in nested form. Different methods for large circuits symbolic analysis mainly relay on hierarchical decomposition and can be classified as graph based [1, 2] and matrix based [3-7] techniques.

The method for symbolic analysis presented in this paper is topology oriented and represents a modification of Topology Decision Diagram method (TDD) developed by the same authors [8]. It generates exact symbolic network function in nested form. The proposed procedure represents symbolic network function by a diagram, like in DDD based algorithm [6, 7].

Graphic representation of the expression allows more efficient symbolic manipulation, derivation and evaluation. Instead of matrix entries in DDD, vertices in TDD are admittances or transconductances. The TDD method will be explained briefly in the next section.

II. TOPOLOGY DECISION DIAGRAM

We consider linear, time invariant, lumped $RLCg_m$ circuits characterized by network function in form of rational functions in the complex frequency s and the circuit parameters p that can be presented by (1) .

$$H(s)=\frac{N(s)}{D(s)}=\frac{\sum_i\left(\prod_j\left(p_{i,j}\cdot s^{n_j}\right)\right)}{\sum_k\left(\prod_l\left(p_{k,l}\cdot s^{n_l}\right)\right)} \tag{1}$$

Authors are with Department of Electronics, Faculty of Electronic Engineering, University of Niš, Aleksandra Medvedeva 14, 18000 Niš, Serbia & Montenegro, E-mail: srdjan@elfak.ni.ac.yu, predrag@elfak.ni.ac.yu.

where n_i is power of complex frequency $n_i \in \{0, 1\}$, while p_{ij} and p_{kl} represent circuit parameters.

Factorization of the network function in arbitrary circuit parameter p_k can be expressed as follows.

$$H_{0pk}=\lim_{p_k\to 0}H=\frac{N(p_k=0)}{D(p_k=0)}=\frac{N_{0pk}}{D_{0pk}} \tag{2}$$

$$H_{\infty pk}=\lim_{p_k\to \infty}H=\frac{N(p_k\to\infty)}{D(p_k\to\infty)}=\frac{N_{\infty p_k}}{D_{\infty p_k}}=\frac{p_k\cdot\frac{dN}{dp_k}}{p_k\cdot\frac{dD}{dp_k}} \tag{3}$$

$$H=\frac{N_{0p_k}+p_k\cdot N_{\infty p_k}}{D_{0p_k}+p_k\cdot D_{\infty p_k}} \tag{4}$$

Numerator and denominator are obtained separately. The expressions of the numerator and denominator of the circuit function are not known in advance, but one can easily determine the corresponding topology reducing circuit by setting $p_k\to 0$ and $p_k\to\infty$. The former case corresponds to elimination of parameter p_k, while $p_k\to\infty$ corresponds to parameter extraction [9]. From the aspect of the circuit topology, the parameter elimination can be treated as a branch reduction and parameter extraction as a node reduction .

During the circuit reduction every two-port device is treated as an admittance. When $RLCg_m$ circuits (CMOS circuits) are in scope, the only other type of devices besides the admittances is voltage controlled current source (VCCS).

When an admittance is the considering parameter, then $p_k\to 0$ corresponds to removed parameter from the circuit. Oppositely, the admittance is replaced by short for $p_k\to\infty$.

If p_k is a transconductance of a VCCS, the circuit topology remains unchanged except in two cases:

1. the current is controlled by the voltage across the same branch and VCCS operates as an admittance;
2. two or more VCCSs, mutually control each other and all transconductances, g_m, are extracted ($g_m=p_k\to\infty$); this implies that nodes of all these VCCSs are connected into one node.

During this recursive process, a tree-like topology decision diagram (TDD)is formed.

Every step in TDD construction results in new circuit topology, strongly related to the former one. During this

1-4244-0117-8/06/$20.00 © 2006 IEEE

process, some of the branches are eliminated and some of the circuit nodes are merged into one. Eventually, TDD outcomes with a graphical representation of the circuit function in nested form.

During the generation of TDD, each vertex in the diagram represents a circuit. The procedure for new vertex generation (child vertex), from previously generated vertex (parent vertex) in TDD can be:

- parameter elimination,
- parameter extraction or
- combination of both.

During the every step of TDD generation, symbolic expressions representing numerator or denominator of the transfer function are divided into two addendums. Both of these expressions do not contain the circuit parameter. They correspond to two new vertices of TDD and simultaneously, to two subcircuits. The first represents the part of the circuit that was independent of that parameter – actually the part from which the parameter was eliminated (branch reduction). The second is obtained after the parameter was extracted (node reduction). Elimination and extraction are always applied together during TDD generation, as graph in Figure 1 shows. These two procedures can be seen as a circuit device suppression with respect to the circuit topology reduction.

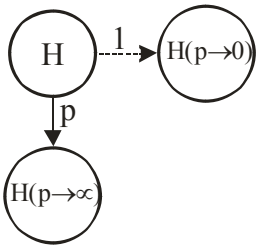

Figure 1. Suppression of a parameter.

The number of product terms (admittance order) in each addendum is the same and it is equal to the number of independent nodes in the network. Elimination of one circuit parameter results in a new vertex at the same level with the outgoing edge value equal to 1. After extraction of a single parameter, the number of nodes is decreased for 1 and admittance order is reduced for one. This corresponds to the lower level node in TDD. Outgoing edge is directed to the child node on the lower level of TDD, with weight p_k^l for k-th parameter extracted at l-th level, k=1,...,m_l), as shown in Figure 2.

The number of addenda at l-th level is equal to the number of eliminated parameters, while admittance order is equal to the number of levels. Actually, the subcircuit obtained at l-th level where m_l parameters are eliminated, is characterized by sum of m_l products each consisted of l multipliers.

Set of l parameters in every of m_l paths from leaf up to the root, represents one addendum in circuit function. Simultaneously, from the scope of circuit topology, it represents the set of branches connecting all nodes in the circuit [10].

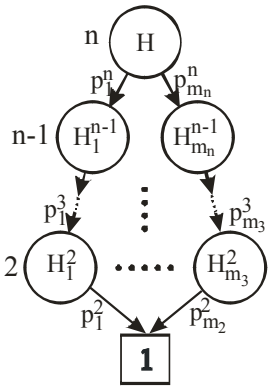

Figure 2. Construction of TDD.

It follows that less TDD nodes will be generated, if one extracts subsequently parameters from the branches connected to common node. Therefore, the fewest parameters set that can be extracted corresponds to the circuit node connecting the smallest number of branches.

In conclusion, the construction of the circuit function expression in symbolic form starts from the lives of TDD and proceeds up to the root, while vertices are represented by symbolic expressions.

The previous analysis indicates that proper ordering during construction of TDD has significant affect on efficiency of TDD method. Therefore the next section describes ordering method that guaranties more compact symbolic expression form.

III. REORDERING

Any part of the circuit can be generally represented as an n-port network, where n is the number of boundary nodes between this subcircuit and the rest of the circuit. The subcircuit is described in terms of the corresponding n-port network parameters . The nodes of the whole circuit can be divided, for the considered subcircuit, into three disjoint groups:

- internal,
- boundary and
- remainder.

We will consider the simplest case where circuit H is divided into subcircuits H_1 and H_2, by tearing node T, as illustrated in Figure 3.

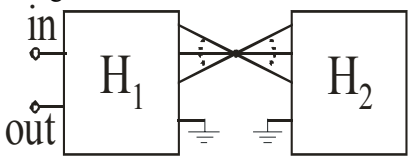

Figure 3. Two subcircuits with one common node.

Suppose that H_1 and H_2 have n_1 and n_2 nodes, respectively. If the controlled generators do not transmit

signal between subcircuits, H_2 can be treated as a two port. In this case, it can be presented as an admittance, according to Figure 4.

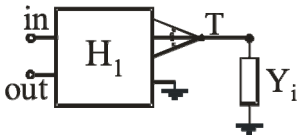

Figure 4. Modeling of subcircuit H_2 by an admittance.

According to (4) the resulting expression of the circuit function is:

$$H = \frac{N_0 + N_1 Y}{D_0 + D_1 Y} = \frac{N_0 N_Y + N_1 N_Y}{D_0 D_Y + D_1 D_Y}, \quad (5)$$

where:

$$Y = \frac{N_Y}{D_Y} \quad (6)$$

It follows that the transfer function is determined by six analytic expressions. Namely, N_0, D_0, N_1, D_1, N_Y, and D_Y. Expressions N_Y and D_Y are obtained from H_2. The reordering method relay on fact that these two expressions share many subexpressions. As will be explained, this can save a lot of computing time and to results in more compact circuit expression in symbolic form.

Let suppress H_2 first. After extraction of n_2 parameters successively (n_2 being the number of internal nodes), topologies obtained as lives in TDD are almost identical. In order to exploit the similarity between circuits, it is advised to proceed with admittance Y_i (i=1,…,m_{n1}) that is connected to the terminal nodes, as indicated in Figure 4.

The number of nodes in the whole circuit is n_1+n_2 where n_1 is the number of nodes in H_1. After n_2 node reductions the remaining number of nodes is equal n_1.

The parameter extraction and elimination from all subcircuits that represent vertex on n_2–th level in TDD, result in two identical circuits. Accordingly, all outgoing edges are directed only to two vertices. The part of the TDD related to this step, is illustrated in Figure 5a and the corresponding circuits are shown in Figure 5b

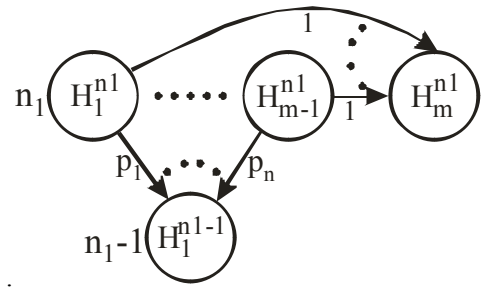

Figure 5. a. Reduction into two vertices.

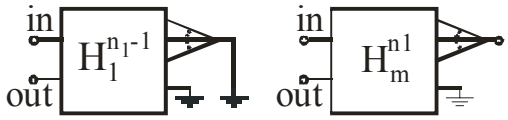

Figure 5. b. Two corresponding circuits.

In the subsequent step, two mutually independent sub-diagrams that correspond to these two circuits are formed. The proposed vertex ordering exploit shearing among expressions N_y and D_y in (6). Instead of two independent subdiagrams, only one is sufficient to provide generation of both expressions.

The order of parameter extraction can be reversed. Namely, it is possible to start with analysis of subcircuit H_1 instead of H_2. More compact expression will be gained if the larger subcircuit is analyzed first because the shearing of common expressions is exploited in more efficient manner.

Generally, bipartiotioning contains more than two tearing nodes. The circuit reduction obtained by extraction/elimination of parameters that belong to one subcircuit will result in many vertices with the identical circuit topology. The difference between circuits that correspond to these vertices are exploited in branches connected between terminal nodes. Extraction and elimination of these parameter will give identical circuits.

The number of levels in TDD corresponds (is equal) to the number of terminal nodes between two subcircuits. Simultaneously, the number of circuits corresponds (is equal) to the combination of the shortcuts and opens between every pair of terminal nodes.

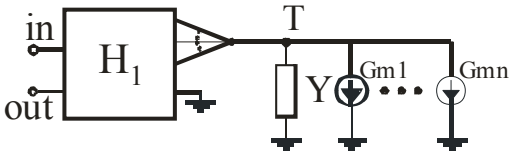

Figure 6. Modeling of a subcircuit by VCCS and admittance.

The procedure can be spread to the subcircuits that contain VCCS controlled by voltages from the rest of the circuit. The necessary condition is that the controlled generators transmit signal in one direction. Namely, from the rest of the circuit to the subcircuit. This case is illustrated in Figure 6.

IV. EXAMPLE

The efficiency of the circuit parameter reordering is illustrated on the ladder network illustrated in Figure 7.

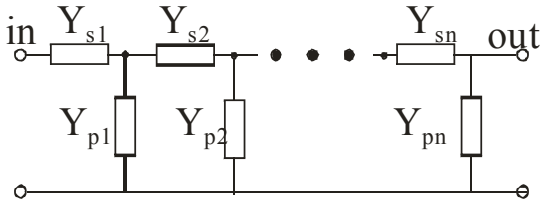

Figure 7. The ladder network.

Table I presents number of multiplications, additions, and intermediate expressions for different number of sections *n*. Obviously, reordering of the circuit

parameter extraction in TDD gives significantly compacted results in terms of number of operations.

TABLE I

n # of sections	without oredering			with ordering		
	#mul	#add	#expr.	#mul	#add	#expr.
5	18	10	33	17	10	15
10	51	34	90	37	24	36
15	94	70	165	62	40	60
20	134	107	246	82	54	81
30	258	212	459	127	84	126
40	332	261	594	172	114	171
50	481	379	870	217	144	216
100	1218	948	2140	442	294	441

Tabele II gives the simulation time for the same example. Simulation is performed on Pentium III processor at 551 MHZ with 256 MB of RAM.

TABLE II

n # of sections	time [mS]	
	without ordering	with ordering
5	13,3	9,7
10	53,57	35,25
15	180	90,8
20	230	157
30	625	401
40	1670	916
50	2070	1350
100	14000	11200

V. CONCLUSION

The primary goal of the symbolic analysis, which result in nested form of network function, is to generate expressions as compact as possible. Hierarchical decomposition is imposed as a natural component of this procedure. Optimization goal for the proposed method is minimization of the vertices number in assigned TDD. The proposed reordering of circuit parameter extraction can significantly reduce the size of TDD that describes circuit function.

Symbolic circuit analysis based on the TDD generation is very suitable for circuit partitioning. Every new TDD node represents a circuit with reduced topology. This paper considers a ladder example simple enough to serve for method explanation but sufficiently complex to explore benefits of the reordering method. The proposed procedure can be spread to the bipartioning with more tearing nodes.

The compactness of symbolic expression is achieved by shearing common expressions between subcircuits having same topology.

ACKNOWLEDGEMENT

This work was partially supported by the Serbian Ministry of Science and Environment Protection through project No. TR 006108.B.

REFERENCES

[1] A. Konczykowska and J. Strzyk, “Computer analysis of large signal flowgraphs by hierarchical decomposition method,” in *Proc. European Conf. Circuit Theory Design*, (Warsaw, Poland) , 1980, pp. 408-413.

[2] Marwan M. Hassoun, Kevin S. McCarville “Symbolic analysis of large-scale networks using a hierarchical signal flowgraph approach” *J.Analog VLSI Signal Process*., vol. 3, pp. 31-42, Jan. 1993.

[3] S.J. Jou, M. F. Perng, C. C. Su and C. K. Wang, "Hierarchical Techniques for symbolic analysis of large electronic circuits" *IEEE Inter. Symp. Circuits and Systems*, pp.21-24, June 1994.

[4] Marwan M. Hassoun, Pen-Min Lin “A hierarchical network approach to symbolic analysis of large-scale networks” *IEEE Trans.Circuits Syst*., vol. 42, pp. 201-211, April 1995.

[5] S. Đorđević, P. Petković, “A hierarchical approach to large circuit symbolic simulation”, *Microelectronics Reliability,* vol.41 , pp. 2041-2049, 2001.

[6] X.-D.Tan and C.-J.Shi, “Hierarchical symbolic analysis of analog integrated circuits via determinant decision diagrams,” *IEEE Trans.Computer-Aided Design*, vol. 19, pp. 401-412, Apr. 2000.

[7] C.-J.Richard Shi, and Xiang-Dong Tan, “Compact Representation and Efficient Generation of s-Expanded Symbolic Network Functions for Computer-Aided Analog Circuit Design,” *IEEE Trans. Computer-Aided Design*, vol. 20, pp. 813-827, Jul. 2001.S. Đorđević and P. M. Petković, ”Generation of Factorized Symbolic Network Function by Circuit Topology Reduction”, *Proceedings of MIEL'04*, Niš, 2004 pp. 773-776.

[8] Đorđević, S., Petković, P., "A Modified Method For Symbolic Network Function Extraction Based On Circuit Topology Decision Diagram", *Proc. of the XLIX Conf. of ETRAN, ETRAN 2005*, Vol. I, pp. 103-106, June 2005, Budva, in Serbian.

[9] Đorđević, S., Petković, P, “Generation of Factorized Symbolic Network Function by Circuit Topology Reduction”, Proceedings of MIEL'04, Niš, 2004 pp. 773-776

[10] L. O. Chua and P. M. Lin, *Computer Aided Analysis of Electronic Circuits-Algorithms and Computational Techniques*. Englewood Cliffs, NJ: Prentice-Hall, 1975.