DIRECT SYMBOLIC ANALYSIS OF LARGE ANALOG NETWORKS

Janusz A. Starzyk and Jun Zou

School of Electrical Engineering and Computer Science
Ohio University, Athens, OH 45701
Email: starzyk@bobcat.ent.ohiou.edu

ABSTRACT

A new symbolic analysis approach for large analog integrated circuits is presented in this paper. It is based on the Coates graph representation and nodes exploding technique for construction of symbolic formulas. It can symbolically analyze large analog ICs without circuit decomposition. There are no limitations for the circuit topology and size. The analysis time and the memory required to store the symbolic formulas are estimated to increase pseudo-linearly with the number of the circuit nodes.

I. INTRODUCTION

As the complexity of analog and mixed signal circuits increased, the need for efficient analog circuit automation design tools became critical. Symbolic analysis is a technique that can provide insight into the circuit behavior by obtaining symbolic formulas which link design parameters, network topology and desired characteristics. Its importance in analog circuit design grew as a result of the success of modern symbolic analyzers, such as ISAAC [1], SCAPP [2], etc.. Symbolic analysis is helpful in development of analytic models and automatic sizing of the circuit cells for given circuit specifications with the use of the optimization programs. It is useful in testability analysis, sensitivity analysis, and fault diagnosis [3].

It is difficult to symbolically analyze large analog networks because of the exponential growth of the number of product terms in symbolic formulas with the circuit size [4]. This difficulty was partially overcome by various approximation [1], [5], and decomposition [6], [7] approaches. The approximation approach improves the interpretability of the results and achieves a balance between expression accuracy and its simplicity. Circuit decomposition allows the symbolic analysis of large analog circuits. Decomposition was thought of as the only hope for symbolic analysis of even larger analog circuits than those that can be analyzed today. But most of the circuits don't possess the loose coupling structure and high regularity desired in the hierarchical decomposition approach. In addition, the decomposition approach and related symbolic formulas are complex and require an advanced knowledge of the graph theory. Therefore, in spite of its development in early eighties [6], [8], its implementation in modern analog simulation tools has been slow.

A new, direct, symbolic analysis approach applicable for large analog circuits is presented in this paper. It is based on the Coates graph representation and nodes exploding technique for construction of symbolic formulas. It can symbolically analyze large analog integrated circuits without circuit decomposition. There are no limitations for the circuit topology and size. The analysis time and the memory required to store the symbolic formulas are estimated to increase pseudo-linearly with the number of the circuit nodes. In addition, the underlying graph theory is simple and the method is easy to program.

The rest of the paper is organized as follows. In section II, the related signal flowgraph representation and node swapping are given. The node exploding technique and optimization of the node exploding order are presented in section III. In section IV, the complete symbolic analysis algorithm is described and is illustrated with an example. Tree manipulation and simplification of the obtained symbolic expression are described in section V. Experimental results are presented in section VI. Finally, the conclusion is given.

II. SIGNAL FLOWGRAPH REPRESENTATION AND NODE SWAPPING

Symbolic analysis is used in linear or linearized networks to obtain network transfer characteristics. The linearized models are used to obtain linearized equations suitable for numerical and symbolic analysis. We assume that the given network has been represented by a linearized equation in which a coefficient matrix and possibly the right hand side vector contain symbolic parameters of the design. Many generalized formulation techniques (like modified nodal, hybrid, etc.) can be used to obtain such equations. Network equations are described by the

Coates flowgraph [9], in which nodes represent signals and the links represent the circuit parameters. Our aim is to express the desired network characteristics by the rational functions of the circuit parameters as described in [9].

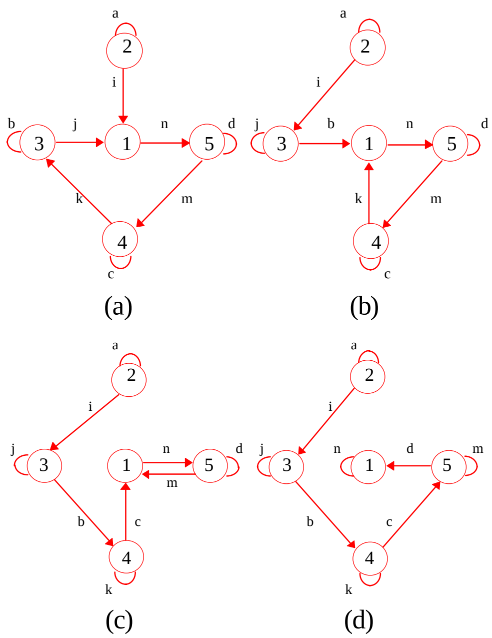

Fig. 1a) Coates graph with singular node, b) Graph after swapping nodes 1 and 3, c) Graph after swapping nodes 1 and 4, d) Graph after swapping nodes 1 and 5

The proposed symbolic analysis technique requires that the analyzed Coates graph has a self-loop at each node except the input nodes (nodes representing the independent signals). A Coates graph which does not have self-loops at all output and internal nodes is called a singular graph. The node swapping technique is used to eliminate the singularity of a singular graph.

Consider a portion of the signal flowgraph shown in Fig.1a). The original graph in this example is singular since node1 has no self-loop. This singularity can be removed by the following procedure:

1. Find the shortest directed loop which contains the singular node (nodes).
2. Swap the singular node with its neighbor which has a link directed to the singular node.
3. While swapping two nodes all links directed towards the first node are transferred to the second node and opposite - all links directed towards the second node are transferred to the first one.
4. The starting nodes of the directed links are unchanged. Swapping two nodes shortens the loop by one node.
5. Repeat 2 and 3 until the singularity is removed.

The transformations of links in order to remove the singularity of node1 in Fig. 1a) are shown in Figs. 1b)-1d). The node swapping algorithm has to be executed before the symbolic analysis of the graph begins.

Simplification of the signal flowgraph before the symbolic analysis is performed may reduce the number of operations needed to obtain the final results and improve the interpretability of the obtained formulas. This simplification is attained by using the superposition of nodes incident to the links of identical parametric values. All elementary flowgraphs of circuit elements presented in Table III of [8] can potentially benefit from this simplification.

III. NODE EXPLODING

The node exploding is the basic technique to obtain the symbolic transfer functions in the analyzed flowgraph. Its purpose is to simplify the graph by eliminating one of its nodes and transferring the electrical properties associated with this node to its neighbors. Since the efficiency of the symbolic analysis technique depends on the node exploding order, the nodes must be renumbered in order to optimize the computational effort.

The procedure implemented in the node exploding is as follows:

1. Consider all pairs of links which have one link incoming to the exploded node and one link going out of it. For each pair of links, a new link is created with the weight equal to the product of the weights of the two combined links.
2. All other links incoming to the end nodes of the new links have their weights multiplied by the negative of the self-loop weight of the exploded node.

Let's see an example shown in Fig. 2a). After exploding node3 and performing the described transformation of the link weights, this graph is simplified to the one shown in Fig. 2b). Since node exploding changes the symbolic weights of its incident nodes only, one can generate either more or less complex symbolic formulas of the analyzed network by manipulating the node exploding order.

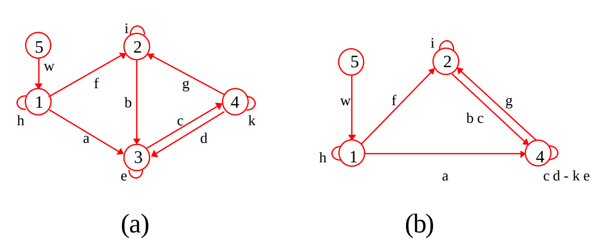

Fig. 2a) The original flowgraph, b) The flowgraph after exploding node3

The node ordering approach adopted in this research is called a minimum distance ordering and is based on the minimum topological distance from a given node to the output node. The minimum distance ordering explodes the nodes which have shortest distance to the output node first. For the single input single output analysis, the input node has the highest number (because it won't be exploded) while the output node the second highest. All internal nodes are numbered accordingly to their minimum distance from the output node. The node numbers in Fig. 2a) are ordered using this node ordering technique. The minimum distance ordering very efficiently reduces the amount of work required in symbolic analysis and can be performed in a linear time.

IV. SYMBOLIC ANALYSIS

The algorithm of symbolic analysis expands symbolic formulas by exploding nodes in a specified order. All start nodes of the new link with lower order than the exploded one are exploded. In order to avoid multiple exploding of the same node at each step of the node removal process, the initial nodes of the newly created links must be searched according to their order.

The signal flowgraph can be represented by a linked list of node structures. Each node structure has three members. The first member is a node number and its self-loop weight s_i. The second one L_{ji} (l_{ji}) describes the symbolic formula (original link weight) of all the links incoming to the node i. The third member S_i describes the symbolic formula of the self-loop of node i. Where i represents the current node and j represents the node coming to node i. These formulas are ordered according to the initial node number of each link. The node structures are linked together according to the node order obtained by node ordering procedure. It is important to differentiate the symbolic formulas from the original link weights which also may connect the specified nodes. These formulas will be subject to modifications resulting from the node exploding and will be composed from either the original link weights or symbolic formulas of the lower order.

In the description of the algorithm which is used to derive the symbolic transfer function we use the following simplified notation:

1. i represents the self-loop weight s_i.
2. $\bar{i}$ represents the symbolic formula S_i.
3. $i_1 \cdots i_k \bar{j}_1 \cdots \bar{j}_m$ denotes the product of the symbolic formulas along the path $i_1 \cdots i_k i$ times the symbolic formulas of the self-loops of the complemented nodes.
4. $\bar{j}_1 \cdots \bar{j}_m$ represents the product of symbolic formulas of the self-loops of the complemented nodes times the original self-loop s_i.

For example, the sequence $1\,2\,3\,\bar{4}\,\bar{5}$ at node 6 denotes the product of $L_{12}L_{23}l_{36}S_4S_5$, and the sequence $\bar{2}\,\bar{3}\,\bar{4}$ at node 5 denotes the product of $s_5S_2S_3S_4$.

The following Table shows all the symbolic manipulations needed to generate the transfer function of the graph shown in Fig. 2a). All sequences in Table are ordered according to the first node number of each sequence.

s_i	S_i	L_{ji} (l_{ji})
1	1	5 5
2	$\bar{1}$	$1;\ 4$ $4\,\bar{1};\ 5\,1$
3	$\bar{1}$ $\bar{1}\,\bar{2}$	$1;\ 2;\ 4$ $2\,\bar{1};\ 4\,\bar{1};\ 5\,1$ $4\,2\,\bar{1};\ 4\,\bar{1}\,\bar{2};\ 5\,1\,\bar{2};\ 5\,2\,\bar{1}$
4	$\bar{3};\ 4\,3$	3 $5\,3$

The first column in the Table represents the first member of the node structure. The generation of the symbolic terms in columns 2 and 3 proceeds sequentially in the specified node order. All the symbolic terms in columns 2 and 3 in the table at node i can be generated from symbolic expressions at nodes of the lower order by performing the following steps:

- step 1: Check the first node j of the first row (l_{ji}) in column 3, if it has higher order than node i, copy node i to the second row of column 2 and the first row of column 3 to its second row, otherwise, explode node j.
- step 2: If the first node j in any sequence of L_{ji} in column 3 has lower order than node i, explode node j.
- step 3: If a first node of L_{ji} in column 3 is node i, move the terms beginning with node i to column 2.
- step 4: Repeat steps 2 and 3 until all start nodes have higher order than node i.

The transfer function T of the example can be obtained as $T = \frac{L_{54}}{S_4} = \frac{5\,3}{\bar{3} + 4\,3}$. Depending on the application, the symbolic formulas can be further expand and simplified.

V. SIMPLIFICATION BY TREE MANIPULATIONS

The symbolic expression obtained directly by the node explosion table is in nested format. It can be represented by a hierarchical tree structure. In the symbolic expression, there are alternate levels of addition and multiplication. We assume that the highest level of operation is multiplication. As a result the entire symbolic expression of the nested formula can be illustrated by the tree shown in Fig. 3. The tree can be simplified at both operation levels. At the multiplication level, if any of the multiplication factors (including sub-trees) are identical, one of them can be removed and the power of another one is increased by one.

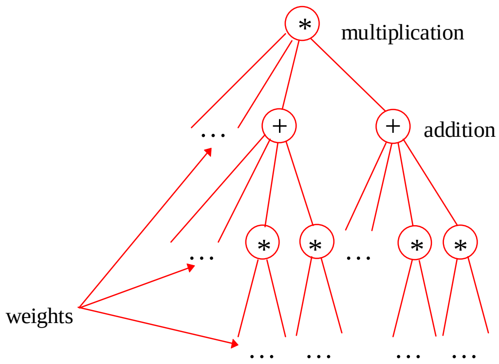

Fig. 3 Tree representation of the nested formula of a symbolic expression

At the addition level, if there are common factors (including sub-trees) of all the addition components, these factors can be removed from current level and placed two levels up (at the next multiplication level). In order to compute common terms, the entire tree and its subtrees are ordered at each level with the single symbols first in alphabetical order, followed by the subtrees.

VI. THE EXPERIMENTAL RESULTS

The above new symbolic analysis technique and the simplification of the symbolic expression by tree manipulation have been programmed in C++. Linked list data structures are used as the basic data structure. A filter circuit shown in Fig. 4) is analyzed as an example.

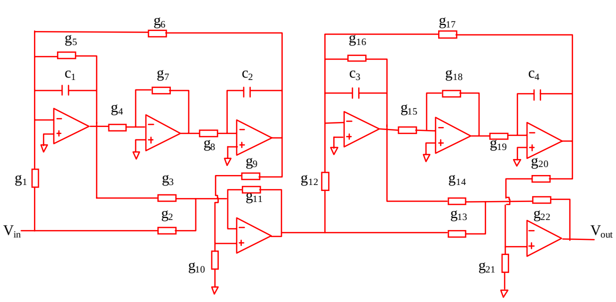

Fig. 4 The example filter circuits

After extraction and combination of the common factors of the original symbolic expression, the exact symbolic formula of the transfer function is obtained as follows:

$$T = \frac{\begin{array}{c} +\,(\,g_1g_4g_8g_9\,(g_{11} + g_2 + g_3) - sc_2g_1g_3g_7\,(\,g_{10} + g_9\,) \\ +\,g_2\,(\,sc_2g_7\,(\,sc_1 + g_5\,) + g_4g_6g_8\,)\ \ (\,g_{10} + g_9\,)\,) \\ (\,g_{12}g_{15}g_{19}g_{20}\,(\,g_{13} + g_{14} + g_{22}\,) - sc_4g_{12}g_{14}g_{18}\,(\,g_{20} + g_{21}\,) \\ +\,g_{13}(sc_4g_{18}(sc_3+g_{16})+g_{15}g_{17}g_{19})(g_{20}+g_{21})) \end{array}}{\begin{array}{c} g_{11}g_{22}\,(\,g_{20} + g_{21}\,)\,(\,g_{10} + g_9\,)\,(\,sc_2g_7\,(\,sc_1 + g_5\,) \\ +\,g_4g_6g_8\,)(\,sc_4g_{18}\,(\,sc_3 + g_{16}\,) + g_{15}g_{17}g_{19}\,) \end{array}}$$

VII. CONCLUSION

Symbolic methods for integrated circuits' analysis and design gained an increasing interest over the past years. The development of new efficient algorithms of the symbolic analysis and their computer implementation were major objectives of this work. In this paper, a new symbolic analysis technique is presented. It is based on the node exploding in the signal flowgraph. It is able to analyze large analog circuits efficiently. The derived symbolic expression can be simplified by tree manipulation.

REFERENCES

[1] G. Gielen and W. Sansen, Symbolic Analysis for Automated Design of Analog Integrated Circuits. Norwell, MA: Kluwer Academic, 1991.

[2] M. Hassoun and P. Lin, “A new network approach to symbolic simulation of large-scale networks,” in Proc. ISCAS, 1989, pp. 806-809.

[3] G. Gielen, P. Wambacq, and W. Sansen, "Symbolic analysis methods and applications for analog circuits: A tutorial overview," Proc. IEEE, vol. 82, pp. 287-304, Feb. 1994.

[4] P. Lin, Symbolic Network Analysis. Amsterdam, The Netherlands: Elsevier, 1991.

[5] P.Wambacq, et al., "Efficient symbolic computation of approximated small-signal characteristics of analog integrated circuits," IEEE J. Solid-State Circuits, vol. 30, no. 3, pp. 327-330, Mar. 1995.

[6] J. Starzyk and A. Konczykowska, "Flowgraph analysis of large electronic networks," IEEE Trans.Circuits and Systems, vol. 33, no. 3, pp. 302-315, Mar. 1986.

[7] M. Hassoun and P. Lin, “A hierarchical network approach to symbolic analysis of large scale networks,” IEEE Trans. Circuits and Systems - I: Fundamemtal Theory and Applications. vol. 42, No. 4, pp. 201-211, April 1995.

[8] J. Starzyk, "Signal flow-graph analysis by decomposition method," IEEE proc. Electronic Circuits Sys., no. 2, pp. 81-86, 1980.

[9] W. Chen, Applied Graph Theory-Graphs and Electrical Networks. Amsterdam, The Netherlands: North-Holland, 1976.