

## Upward Topological Analysis of Large Circuits Using Directed Graph Representation

J. A. STARZYK AND E. SLIWA

**Abstract**—This paper presents the method of topological analysis of large LLS networks with the use of hierarchical decomposition of the network graph. It is assumed that the network is represented by a directed graph.

An algorithm of upward hierarchical analysis of a partitioned graph is presented. The algorithm allows symbolic analysis of large networks with the number of elements kept as symbols practically unlimited. The computational time linearly depends on the network size. A computer program using techniques described is also presented.

### I. INTRODUCTION

The major disadvantage of topological methods of symbolic analysis lies in the rapid increase of the number of terms in topological formulas (approximately exponential w.r.t. number of network nodes) [1]. This has practically limited their application to symbolic hand analysis of very small circuits. Computer programs based on direct topological analysis can handle in practice networks having up to 10–15 nodes only.

To overcome computational difficulties a direct decomposition method (Chen [2] for  $n$ -vertex bisection, and Konczykowska and Starzyk [3] for the general case) was developed. This method enables us to analyze networks having up to 30 nodes (program ADEN [4] written for nullator–norator representation). The time of analysis for this method increases exponentially with the square root of the number of network nodes.

A significant improvement was achieved when the method of analysis by hierarchical decomposition [5] was introduced. Program HADEN [6] based on so-called downward hierarchical analysis, makes analysis of networks having more than 100 nodes feasible, and the algorithm used is polynomially bounded.

In this paper an upward hierarchical analysis method is presented. The method allows practically all symbolic analysis of quite large networks with computational time linearly dependent on the network size.

### II. TOPOLOGICAL REPRESENTATION

Here we briefly recall some basic notions which will be subsequently used in the paper (see also, for example [5]). We shall concentrate on topological methods which use a weighted, directed graph as a network model, i.e., Mason unistor graph and Chen digraph [1]. Because the later representation can be obtained simply by changing directions of the edges of the first one, we are concerned with the unistor graphs only. Symbolic network functions are obtained by enumeration of multitrees of various types of the network graph [1].

**Definition:** A directed  $k$  tree  $t_V$  is simply a  $k$  tree of graph  $G$  with set of components  $V$  given by

$$V = \left\{ (r_1, v_1^1, \dots, v_{m_1}^1), \dots, (r_k, v_1^k, \dots, v_{m_k}^k) \right\} \quad (1)$$

and the  $i$ th component of this  $k$  tree,  $i=1, \dots, k$ , contains vertices  $r_i, v_1^i, \dots, v_{m_i}^i$  where  $r_i$  is the reference vertex of the

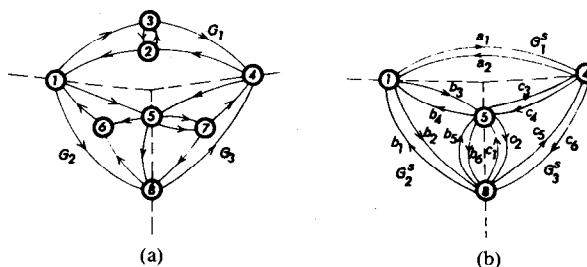


Fig. 1. Graph  $G$  and its trisection (a) and substitute graph of decomposition  $G^s$  (b).

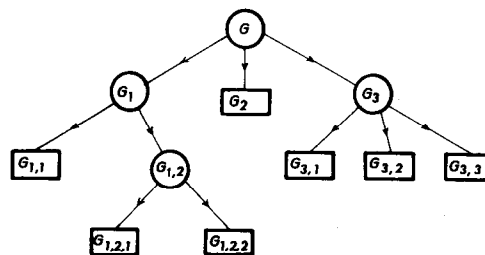


Fig. 2. Example of the tree of decomposition.

component. By a weight  $|t|$  of a  $k$  tree  $t$  we mean

$$|t| = \prod_{e \in t} y_e \quad (2)$$

where  $y_e$  is the weight of edge  $e$  of the  $k$  tree  $t$ . For a  $k$  tree consisting of isolated vertices only we assume  $|t| = 1$ .

A weight function  $|T|$  of a set of  $k$  trees  $T$  is defined as

$$|T| = \sum_{t \in T} |t|. \quad (3)$$

Let graph  $G(X, E)$  be a directed graph which represents a lumped, linear stationary (LLS) circuit. Let us define the set  $B$  of block vertices of  $G$  as a set of terminal vertices (for a two-port network there are four such vertices). Let us suppose that  $G$  has been partitioned onto  $l$  subgraphs  $G_i(X_i, E_i)$ ,  $E_i \cap E_j = \emptyset$  for  $i \neq j$ ,  $\cup_{i=1}^l E_i = E$  (only vertex decomposition is considered in this paper; generalization for edge and mixed decomposition follows immediately). Let  $S = \{G_1, \dots, G_l\}$ . By block vertices  $B_i$  of  $G_i$  we mean the vertices of  $B$  and cut vertices, which belong to  $X_i$ . For every graph  $G_i$  we define the substitute graph  $G_i^s(B_i, E_i^s)$  as the complete graph spanned over block vertices  $B_i$ . Union of all the substitute graphs  $G_i^s$  forms the substitute graph of decomposition  $G^s$  (we assume that  $G^s$  has the same set of block vertices as  $G$ ). An example of decomposition and its substitute graph is shown in Fig. 1.

For substitute graphs we can introduce another useful notion.

**Definition:** A  $k$  tree  $t_V$  of a substitute graph  $G^s$  is called a proper  $k$  tree if there is no directed path of the length greater than one in any subgraph  $t_V \cap E_i^s$ .

For example, in Fig. 1 the tree  $t_1' = \{b_4, c_1, c_4\}$  is a proper tree, while the tree  $t_1'' = \{b_4, b_5, c_4\}$  is not because  $t_1'' \cap E_2^s = \{b_4, b_5\}$  which is a directed path of length 2.

Any graph  $G_i$  which is too large to be analyzed directly may be further partitioned onto graphs  $G_{i,j}$ ,  $j=1, \dots, l_i$ , and a substitute graph for this decomposition can be set up. Such a procedure, called hierarchical decomposition, can be continued until sufficiently small graphs are obtained. The process of decomposition can be illustrated by the so-called tree of decomposi-

Manuscript received November 16, 1981; revised July 28, 1982.  
E. Sliwa is with the Institute of Electronics Fundamentals, Technical University of Warsaw, 00-665 Warsaw, Poland.

J. A. Starzyk is with the Department of Electrical and Computer Engineering, Ohio University, Athens, OH 45701, on leave from the Institute of Electronics Fundamentals, Technical University of Warsaw, Warsaw, Poland.

TABLE I  
UNISTOR MODELS OF NETWORK ELEMENTS

Element	Autonomic FUM	Nonautonomic FUM
VVT 		
CCT 		
CVT 		
Opamp 		

$a_{ij}$  by an unistor directed from vertex  $j$  to vertex  $i$  with weight  $-a_{ij}$ ,

- ii) by replacing all elements by their unistor models.

The unistor models of an element can be obtained directly from IAM of the element. If an element has no admittance description, the so-called *formal unistor model* (FUM) can be introduced. Table I shows examples of some autonomic and nonautonomic FUM's. These models were obtained on the basis of modified nodal equations of elements.

Autonomic FUM's do not change the structure of a graph incident with them, while nonautonomic ones have fewer edges and vertices. The graph which contains nonautonomic FUM's has to be transformed in the following way:

- i) any edge outgoing from a vertex of the nullator is replaced by two edges as in the Fig. 3(a),
- ii) any edge incoming to a vertex of norator is replaced by two edges as in the Fig. 3(b),
- iii) vertices of zero incoming degree are identified with corresponding vertices of zero outgoing degree and all self loops are removed.

Having the unistor graph of a given network, topological analysis of it can be performed. Formulas for basic network functions of a two-port network can be found, for example, in [1].

III. GRAPH DECOMPOSITION

The unistor graph of an electronic network should be hierarchically decomposed before running the analysis procedure. The decomposition of the network graph should satisfy the following conditions:

- i) Successive partitions should introduce as few new block vertices as possible (because the number of block vertices strongly influences time of computation and computer storage used),
- ii) proper blocks should be of the size near optimum (a graph should be partitioned only if it is worthwhile to do so).

The decomposition can be made manually or automatically. A heuristic algorithm satisfying these conditions was proposed in [7] and implemented in a computer program is described in Section VI. The time of partition linearly depends on the number of graph nodes.

IV. ANALYSIS OF DECOMPOSED GRAPH

In this section we will present the method and the algorithm for analysis of decomposed graph and its subgraphs.

Theorem 1

The set  $T_V$  of all multitrees  $t_V$  of a decomposed graph is given by

$$T_V = \sum_{l \in Q_V} \prod_{G_i \in S} T_{V_i}^l \tag{4}$$

where  $Q_V$  is the set of all proper  $k$  trees of the substitute graph of decomposition  $G^s$ .  $T_{V_i}^l$  is the set of all  $k_i$  trees of graph  $G_i$  having set of components  $V_i$ .  $V_i$  is determined as follows. Let  $G_i^l \triangleq G_i(B_i, E_i^s)$ , and let  $k_i$  be the number of components of  $G_i^l$ , where each of these components is of the form of an incoming star, following the definition of a proper  $k_i$  tree. The  $j$ th component of  $G_i^l$  determines a component of  $V_i$  of the form  $(r_j, v_1^j, \dots, v_m^j)$ , where  $r_j$  is the reference node of that component, and  $V_i^l, l=1, \dots, m$  are remaining vertices of that component.

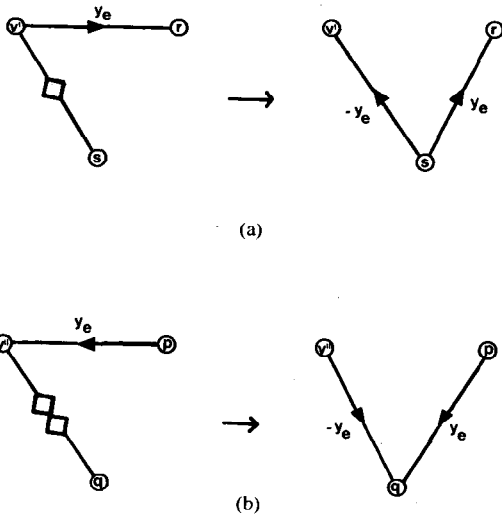


Fig. 3. Transformation of edges outgoing from nullator (a), and incoming to norator (b).

tion. Vertices of this tree are assigned to graphs (both graphs which were partitioned and *proper blocks*—i.e., unpartitioned parts of network graph). Fig. 2 shows an example of such a tree. Graph  $G$  was partitioned first onto  $G_1, G_2$ , and  $G_3$ ;  $G_1$  and  $G_3$  were further partitioned:  $G_3$  onto  $G_{3,1}, G_{3,2}, G_{3,3}$ , and  $G_1$  onto  $G_{1,1}$  and  $G_{1,2}$ .  $G_{1,2}$  was finally partitioned onto  $G_{1,2,1}$  and  $G_{1,2,2}$ .

- i) From the *indefinite admittance matrix* (IAM)  $A_a = [a_{ij}]$  by representing every nondiagonal and nonzero element

This theorem shows how the set  $T_V$  of all multitrees  $t_V$  of  $G$  can be obtained on the basis of the sets of multitrees of the subgraphs  $G_i$  and the structure of interconnections of these subgraph, i.e., on the substitute graph of decomposition  $G^S$ .

#### Remark

Equation (4) can be used recursively in the case any of graphs  $G_i \in S$  was further partitioned. Direct application of Theorem 1 and Remark for hierarchically partitioned graphs leads to the algorithm of downward hierarchical analysis described in [5]. Better results can be obtained by means of upward analysis. This is described in what follows.

**Definition:** By the set  $P_B$  of multitrees of a graph  $G$  spanned over the set  $B$  of block vertices we mean union

$$P_B \triangleq \bigcup_V T_V$$

where  $T_V$  is the set of all  $k$  trees  $t_V$  of  $G$  satisfying the condition

$$B = \{r_1, v_1^1, \dots, v_{m_1}^1, \dots, r_k, v_1^k, \dots, v_{m_k}^k\} \quad (5)$$

and summation is taken over all possible  $V$  satisfying (5).

The following corollary follows immediately from Theorem 1 (cf. [8]).

#### Corollary

$$P_B = \sum_{i \in R_B} \prod_{G_i \in S} T_V^i \quad (6)$$

where  $R_B$  is the set of proper multitrees of graph  $G^S$  spanned over  $B$ .  $T_V^i$  is as described in Theorem 1.

Note that  $T_V^i$  is a subset of  $P_B^i$ , where  $P_B^i$  is the set of all multitrees of  $G_i$  spanned over  $B_i$ . In conclusion, (6) allows us to determine the set  $P_B$  of graph  $G$  on the basis of the sets  $P_B^i$  of graphs  $G_i \in S$  and on the structure of their interconnections  $G^S$ .  $P_B$  contains all information about the graph considered as a  $n$ -pole network, where  $n = \bar{B}$ , and will be called a *description* of  $G$ .  $T_V$  for a given  $V$  satisfying (5) will be called a *record* of  $P_B$ . The Remark can be applied to (6). Note that descriptions of subgraphs of  $G$  have to be known before the description of  $G$  can be determined.

Network functions can be easily obtained and expressed as quotients of appropriate subsets of  $P_B$ . For example, for unistor graph of two-port network the voltage transmittance is given by (compare [1])

$$k_u = \frac{|T_{(pr),(qs)}| - |T_{(ps),(qr)}|}{|T_{(r),(spq)}| + |T_{(rp),(sq)}| + |T_{(rq),(sp)}| + |T_{(rpq),(s)}|}$$

Since  $B = \{r, p, s, q\}$ , all sets:  $T_{(pr),(qs)}$ ,  $T_{(ps),(qr)}$ ,  $T_{(rp),(sq)}$ ,  $T_{(rq),(sp)}$ ,  $T_{(r),(spq)}$ ,  $T_{(rpq),(s)}$  are records of  $P_B$ . Similar formulas for other network functions can be easily derived.

Suppose, that decomposition has resulted in  $b$  proper blocks and assume for simplicity the case of hierarchical bisection. The number of substitute graphs will be then  $b - 1$ , and the time of analysis  $\tau$  can be estimated as

$$\tau = b \cdot \tau_p + (b - 1) \cdot \tau_s \approx b(\tau_p + \tau_s) \quad (7)$$

where  $\tau_p$  and  $\tau_s$  are average values of time needed for enumeration of multitrees of proper block and substitute graph, respectively. Provided that the graph has been partitioned onto blocks of similar size, (7) shows linear dependence on network size. Of course it depends on the size of the proper blocks and/or substitute graphs, since both  $\tau_p$  and  $\tau_s$  are exponential functions of their nodes. Similar dependence for downward analysis was proportional to  $n^\alpha$ , where  $n$  is number of network nodes and  $\alpha \approx 2 - 3$  (cf. [5]).

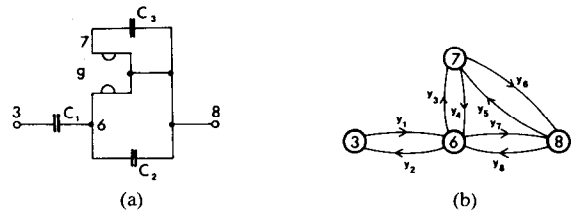


Fig. 4. Active one-port (a), and its unistor graph (b).

TABLE II

Type	Function
$T_{(3)}$	$y_4 y_2 (y_5 + y_8) + y_6 y_2 y_8$
$T_{(8)}$	$y_4 y_7 y_1 + y_6 (y_3 + y_7) y_1$
$T_{(3),(8)}$	$y_4 y_2 + y_4 y_7 + y_6 y_2 + y_6 (y_3 + y_7)$

## V. ENUMERATION OF MULTITREES

There are many efficient multitrees enumerating algorithms elaborated for the purpose of direct topological analysis (cf. [9] and its references). In the upward analysis of decomposed graph we have to enumerate the sets of all multitrees spanned over block vertices for all proper blocks and the sets of all proper multitrees spanned over block vertices for substitute graphs. In this section, we propose an efficient algorithm, based on a well-known remark that the set of directed trees containing edge  $e$  of a graph  $G$  can be obtained by enumeration of all directed trees of  $G_1$ , where  $G_1$  is obtained from  $G$  by short-circuiting  $e$ . Similarly, the set of trees of  $G$  not containing  $e$  is the same as the set of trees of  $G_2$ , obtained from  $G$  by removing  $e$ .

Let us consider graph  $G(V, E)$  having  $n$  vertices. Let  $B$  be the set of block vertices of  $G$  and  $\bar{B} = n_B$ . Vertices of  $G$  can be ordered as follows:

$$h_1, h_2, \dots, h_\epsilon, v_1, v_2, \dots, v_{n_B} \quad (8)$$

where  $h_i \in V - B, i = 1, \dots, \epsilon, v_i \in B, i = 1, \dots, n_B$ .

**Definition:** The *structural matrix*  $M^{(1)} = [m_{ij}]$  of  $G$  is  $n \times n$  matrix, where  $m_{ij}$  is the set of edges outgoing from vertex  $i$  and incoming to  $j$ ;  $m_{ii} \triangleq [0]$ . Rows and columns of  $M^{(1)}$  are enumerated according to (8).

#### Theorem 2

The set of all multitrees of  $G$  spanned over  $B$  can be obtained without duplications by iterative expansion of function  $T(M^{(1)})$ , where

$$T(M^{(i)}) = \bigcup_{j \in J_i} m_{ij} \times T(M_j^{(i+1)})$$

$$T(M^{(n+1)}) = [0]$$

where  $[0]$  denotes the unit element and  $0$  zero element of Wang [10] algebra, respectively,  $m_{ij} \in M^{(i)}, i = 1, \dots, n$  and

$$J_i = \begin{cases} \{i+1, \dots, n\}, & \text{for } i \leq \epsilon \\ \{n - n_B + 1, \dots, n\}, & \text{for } i > \epsilon \end{cases}$$

$M_j^{(i+1)}$  is a matrix obtained from  $M^{(i)}$  after adding the  $i$ th column of  $M^{(i)}$  to the  $j$ th column, and setting the  $i$ th column to zero (for  $i \neq j$ ; for  $i = j, M_j^{(i+1)} = M^{(i)}$ ). While adding we assume that  $m_{ij} + [0] = [0]$ . Extracting  $m_{ii}$  denotes, that new component with reference node  $i$  appeared. The reader should notice

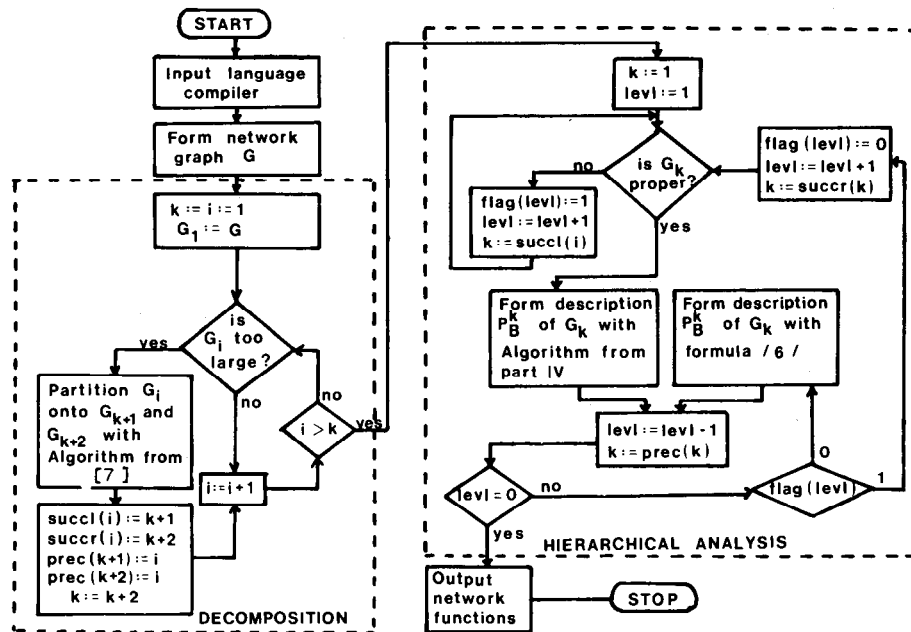


Fig. 5. Flowchart of HADEN2.

that direct application of the Wang algebra to enumeration of multitrees is not efficient because of duplications, as discussed in [2]. To illustrate enumeration of multitrees with the help of the discussed method let us consider the example of active one-port.

#### Example 1

Unistor graph  $G$  of the linear network of Fig. 4(a) is shown in Fig. 4(b) where  $y_1 = y_2 = sC_1$ ,  $y_3 = -y_4 = g$ ,  $y_5 = -g + sC_3$ ,  $y_6 = g + sC_3$ ,  $y_7 = -g + sC_2$ ,  $y_8 = g + sC_2$ . The structural matrix is as follows:

$$M^{(1)} = \begin{matrix} & \begin{matrix} 7 & 6 & 3 & 8 \end{matrix} \\ \begin{matrix} 7 \\ 6 \\ 3 \\ 8 \end{matrix} & \begin{bmatrix} [0] & [4] & 0 & [6] \\ [3] & [0] & [2] & [7] \\ 0 & [1] & [0] & 0 \\ [5] & [8] & 0 & [0] \end{bmatrix} \end{matrix}$$

To obtain all multitrees of  $G$  spanned over  $B$  we use iterative formula from Theorem 2

$$T(M^{(1)}) = [4] \times T \left( \begin{bmatrix} [0] & [2] & [7] \\ [1] & [0] & 0 \\ [5,8] & 0 & [0] \end{bmatrix} \right) + [6] \times T \left( \begin{bmatrix} [0], & [2] & [3,7] \\ [1] & [0] & 0 \\ [8] & 0 & [0] \end{bmatrix} \right)$$

For the sake of simpler notation we will remove  $T(\ )$  and  $x$ ,

$$\begin{aligned} T(M^{(1)}) &= [4] [2] \begin{bmatrix} [0] & 0 \\ [5,8] & [0] \end{bmatrix} + [4] [7] \begin{bmatrix} [0] & [1] \\ 0 & [0] \end{bmatrix} \\ &+ [6] [2] \begin{bmatrix} [0] & 0 \\ [8] & [0] \end{bmatrix} + [6] [3,7] \begin{bmatrix} [0] & [1] \\ 0 & [0] \end{bmatrix} \\ &= [4] [2] [0] [5,8] + [4] [2] [0] [0] \\ &+ [4] [7] [0] [0] + [4] [7] [1] [0] \\ &+ [6] [2] [0] [8] + [6] [2] [0] [0] \\ &+ [6] [3,7] [0] [0] + [6] [3,7] [1] [0]. \end{aligned}$$

On the basis of obtained multitrees we can write weight functions completely describing given one-port (see Table II). It can be now easily analyzed as a part of bigger network, using upward hierarchical analysis.

#### VI. COMPUTER PROGRAM

In this section we give some remarks on the computer implementation of the upward analysis algorithm. The essential problem lies in data structures because of the necessity of storing all sets of multitrees in the computer memory. Any multitree weight  $|t|$  can be written as

$$|t| = C \cdot s^M \cdot \prod_{i=1}^N y_i^{p_i} \quad (9)$$

where  $C$  is product of all nonsymbolic edge admittances of the multitree  $t$ ,  $y_i$  are symbols,  $p_i$  are exponents of symbols (we allow identifying some network elements), and  $N$  is number of symbols.  $s$  is complex frequency,  $M$  is the sum of powers of  $s$  of all edge admittances of  $t$ . Because  $M$  and  $p_i$  are small integers, they may be stored in a single computer word in packed form. In addition, this word can be used as a key to sort multitrees in order convenient for further processing.

The set  $P_B$  of multitrees spanned over  $B$  is the set of sets  $T_\gamma$ . For example, a graph with three block vertices:  $B = \{a, b, c\}$  has description  $P_B$  consisting of ten records

$$T_{(abc)}, T_{(bac)}, T_{(cab)}, T_{(a),(bc)}, T_{(ac),(b)}, T_{(a),(cb)}, T_{(ab),(c)}, T_{(b),(ca)}, T_{(ba),(c)}, T_{(a),(b),(c)}$$

This remark remains true for the set  $R_B$  of proper multitrees of  $G^s$  spanned over  $B$ . The proper multitree of  $G^s$  may be considered as  $l$ -tuple of symbolic addresses pointing to appropriate records of descriptions  $P_B^i$  of subgraphs  $G_i$ ,  $i = 1, \dots, l$ , of graph  $G$ . These addresses can be evaluated during analysis of  $G^s$  and stored in the place of proper  $k$  tree edges.

Final symbolic formula can be interpreted as a sum of products with hierarchical parenthesis

$$\sum_{Q_1} \prod_{S_1} \left( \dots \left( \sum_{Q_k} \prod_{S_k} \left( \sum_{t \in T} \prod_{e \in t} y_e \right) \dots \right) \right) \quad (10)$$

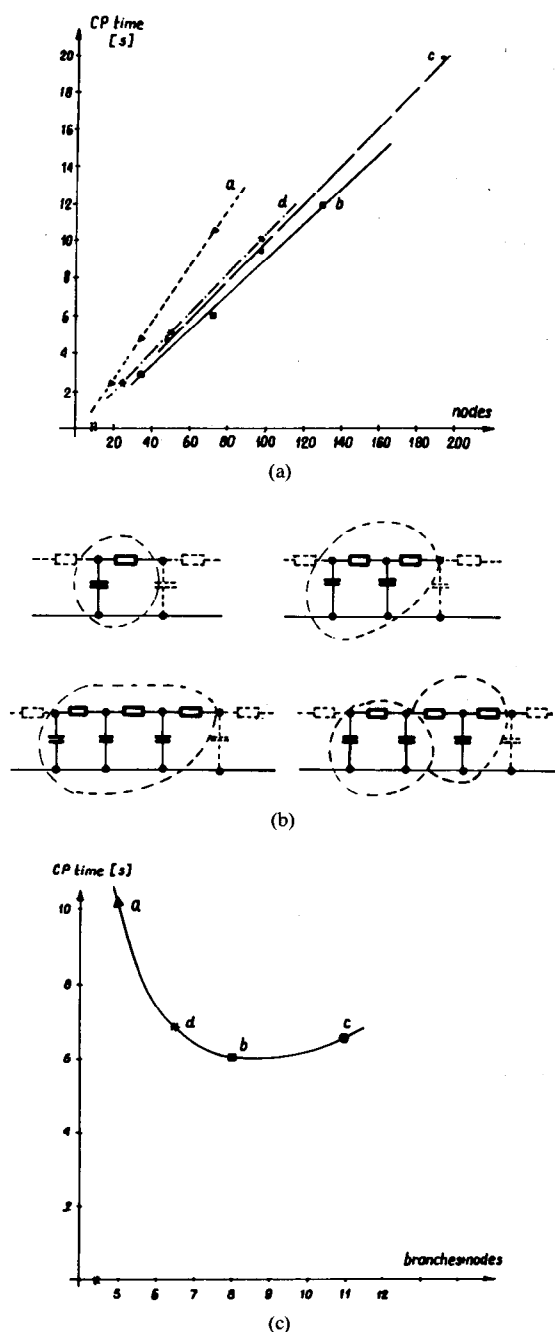


Fig. 6. (a) Dependency of time of analysis w.r.t. the number of network nodes. (b) Proper blocks of ladder decomposition. (c) Dependency of time of analysis w.r.t. the size of proper blocks.

Computing numerical values from (10) can be done without expanding the formula, because intermediate results are stored, thus saving computer time. The form (10) is also convenient for sensitivity analysis because parameters can be easily isolated (since any one of them appears in one proper block only).

A computer program HADEN2 has been written on the basis of the presented algorithms. The program assumes the hierarchical bisection case only. This does not cause loss of generality because any hierarchical decomposition can be presented as hierarchical bisection with the same proper blocks. The flowchart of the program is shown in Fig. 5.

The program was tested for networks having up to 200 nodes. Fig. 6(a) shows the dependence of the time of analysis w.r.t the number of network nodes on the computer CDC-CYBER73. It confirms the predicted linear relationship.

The RC ladder was chosen as test network due to its regular structure. Half of the elements were kept in symbolic form. Tests were performed for partitions resulting in proper blocks as shown in Fig. 6(b). Fig. 6(c) shows how the time of analysis depends on the size of proper blocks of the network. For large proper blocks, the time of their analysis dominates over the time of analysis of interconnections. If proper blocks are too small, the time of analysis of their interconnections increases.

The time of a single graph bisection depends linearly on the graph size. Thus the time of hierarchical bisection is proportional to  $n \log_2 n$ , when  $n$  is the number of network nodes. The time of decomposition for networks having up to 200 nodes is much smaller than the time of hierarchical analysis.

## VII. CONCLUSIONS

This paper presents a method for topological analysis of large linear networks which are represented by unistor graphs. The all-symbolic analysis of large networks is efficiently realized by this method. The analysis time and memory requirements linearly increase with the network size. The method was programmed and computational results are in complete agreement with the theory. Using this method topological analysis can be applied to a various circuit design problems where it was previously impossible.

## REFERENCES

- [1] W. K. Chen, *Applied Graph Theory-Graphs and Electrical Networks*. Amsterdam, The Netherlands: North-Holland, 1976.
- [2] W. K. Chen, "Unified theory on the generation of trees of a graph," *Int. J. Electron.*, "Part I. The Wang algebra formulation," vol. 27, no. 2, 1969, "Part II. The matrix formulation," vol. 27, no. 4, 1969, "Part III. Decomposition and elementary transformations," vol. 31, no. 4, 1971.
- [3] A. Konczykowska and J. Starzyk, "Determination of structural number of partitioned graph," *Arch. Elektrotech.*, vol. z.2, pp. 245-262, 1975, in Polish.
- [4] A. Konczykowska, "Computer program for analysis of decomposed networks," *Proc. European Conf. Circuit Theory and Design (Genova)*, 1976.
- [5] J. Starzyk and E. Sliwa, "Hierarchic decomposition method for the topological analysis of electronic networks," *Int. J. Circuit Theory Appl.*, vol. 8, pp. 407-417, 1980.
- [6] G. Centkowski, J. Starzyk, and E. Sliwa, "Computer implementation of topological method in the analysis of large networks," in *Proc. European Conf. Circuit Theory and Design (Warsaw)*, 1980, pp. 69-74.
- [7] J. Starzyk, "An efficient cluster algorithm," in *Proc. of 5th Czech-Polish Workshop on Circuit Theory*, 1980.
- [8] G. Centkowski, J. Starzyk and E. Sliwa, "Symbolic analysis of large LLS networks by means of upward hierarchical analysis," in *Proc. European Conf. Circuit Theory and Design (The Hague)*, 1981, pp. 328-361.
- [9] S. D. Shieu and S. P. Chan, "Topological formulation of symbolic network functions and sensitivity analysis of active networks," *IEEE Trans. Circuits Syst.*, vol. CAS-21, pp. 39-45, 1974.
- [10] K. T. Wang, "On a new method for the analysis of electrical networks," *Nat. Res. Inst. for Engineering, Academia Sinica Memoir*, no. 2, 1934.