No. SOC-273

UPWARD TOPOLOGICAL ANALYSIS OF LARGE CIRCUITS USING DIRECTED GRAPH REPRESENTATION

J.A. Starzyk and E. Sliwa

November 1981

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. A new approach, using Coates signal-flow graphs, to the element modelling is described.

An algorithm of upward hierarchical analysis of 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.

E. Sliwa is with and J.A. Starzyk is on leave from, the Institute of Electronics Fundamentals, Technical University of Warsaw, 00-665 Warsaw, Poland.

J.A. Starzyk is with the Group on Simulation, Optimization and Control and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

I. INTRODUCTION

Among many methods of analysis of linear electronic circuits topological methods are of particular interest because of the following advantages:

(i) network functions can be obtained in symbolic form which is convenient for further processing, e.g., optimisation or statistical analysis,

(ii) accuracy gained is better than in most of other methods.

The major disadvantage of these methods, which has practically limited their application to symbolic hand analysis of very small circuits, lies in rapid increase of the number of terms in topological formulas (approximately exponential w.r.t. number of network nodes) [1]. 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 elaborated. This method enables us to analyse networks having up to 30 nodes (program ADEN [4] written for nullator-norator representation). Time of analysis, as determined for this method, increases exponentially with the square root of the number of network nodes. However, the method imposes some restrictions on the network structure and can hardly be considered as convenient for the user.

A turning point has been 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 possible. Moreover, time of analysis for the

algorithm implemented is bounded by a polynomial of order, for typical network structures, 2 to 3. This makes it competitive to other numerical methods. However, the method has two major drawbacks:

(i) the number of elements kept in symbolic form is limited to a very small number because of the rapid increase of storage requirements (for example, HADEN evaluates network functions as rational functions of complex frequency s only),

(ii) some parts of the network have to be analysed repeatedly.

The method presented in this paper, called upward hierarchical analysis, enables us to overcome these drawbacks. It allows practically all symbolic analysis of quite large networks having more than 100 nodes.

II. TERMINOLOGY

Here we briefly recall some basic notions which will be subsequently used in the paper (see also, for example [5]). We shall concentrate on such topological methods, which use a weighted, directed graph as a network model. Symbolic network functions are obtained by enumeration of multitrees of various types of the network graph [1].

Definition. By a directed k-tree t_V (simply a k-tree) of graph G with set of components V, where V is of the form

$$V = \{(r_1, v_1^1, \ldots, v_{m_1}^1), \ldots, (r_k, v_1^k, \ldots, v_{m_k}^k)\} \qquad (1)$$

we mean a subgraph of G having the following properties:

1) it contains no loops and n-k edges, where n is the number of graph vertices,

2) i-th component of t_V contains vertices r_i, v_i^i, ..., $v_{m_i}^i$ with r_i being the reference node of the component (i.e., for any vertex v of the component, there exists a directed path from v to r_i).

By a weight $|t|$ of a k-tree t we mean

$$|t| = \prod_{e \in t} y_e \tag{2}$$

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

$$|T| = \sum_{t \in T} |t| \tag{3}$$

Let graph G(V,E) be a directed graph which represents a 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 $G_i(V_i, E_i)$, $E_i \cap E_j = \emptyset$, $\bigcup_i E_i = E$ (only vertex decomposition is considered in this paper; generalization for edge and mixed decomposition follows immediately). Let $S = \{G_1, \ldots, G_\ell\}$. By block vertices B_i of G_i we mean the vertices of B and cut vertices, which belong to V_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 1 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 $t_1'' \cap E_2 = \{b_4, b_5\}$ which is a directed path of length 2.

Any of graphs G_i, which is too large to be analysed directly, may be further partitioned onto graphs $G_{i,j}$, $j = 1, \ldots, \ell_i$, 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 so-called the tree of decomposition. Vertices of this tree are assigned to graphs (both substitute graphs of decomposition 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 $G_{1,2,2}$.

III. ELEMENT MODELS

One of possible ways of solving the system of linear equations

$$\underset{\sim}{A}\ \underset{\sim}{x} = \underset{\sim}{F} \tag{4}$$

is to evaluate the determinant and cofactors of the matrix $\underset{\sim}{A}$. If, for example, (4) is the set of node equations and $\underset{\sim}{A}$ is admittance matrix, we can write

$$\underset{\sim}{A} = \underset{\sim}{\lambda}_1\ \underset{\sim}{Y}_b\ \underset{\sim}{\lambda}_2^T \tag{5}$$

where $\underset{\sim}{\lambda}_1$ is basis incidence matrix of network unistor graph, $\underset{\sim}{\lambda}_2$ is obtained from $\underset{\sim}{\lambda}_1$ after setting all elements -1 to 0. $\underset{\sim}{Y}_b$ is diagonal matrix of edge admittances.

The determinant of $\underset{\sim}{A}$ is equal to the weight function of the set of all directed trees of the unistor graph [1]. Similar formulas for the cofactors of $\underset{\sim}{A}$ are known [1].

If in (5) $\underset{\sim}{\lambda}_1$ is replaced by incidence matrix $\underset{\sim}{\Lambda}_1$, the <u>indefinite admittance matrix</u> (IAM) is obtained

$$\underset{\sim}{A}_a = \underset{\sim}{\Lambda}_1 \underset{\sim}{Y}_b \underset{\sim}{\Lambda}_2^T \tag{6}$$

Similar formulas are known for other topological representations of the network (e.g. for Chen digraph, signal flow graphs, etc). In the paper we shall consider only directed graphs, i.e., Mason unistor graph and Chen digraph [1]. Because the first representation can be obtained simply by changing directions of the edges of the second one, we shall further concern the unistor graphs only. Edges of such graphs, called unistors, can be interpreted as current sources controlled by one of node voltages.

A network unistor graph can be obtained in two ways:

(i) from IAM $\underset{\sim}{A}_a = [a_{ij}]$ by representing every nondiagonal and nonzero element 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, so-called a <u>formal unistor model</u> (FUM) can be introduced. Having element equations, signal-flow graph (SFG) of the element can be obtained. Vertices of that SFG represent element variables: node voltages and optionally branch currents or voltages. By replacing edges of the SFG of an element (edges of SFG will be called <u>transitors</u>) by appropriate set of unistors we obtain autonomic FUM of the element. Unistors of

such a model are called formal because they have different interpretation from normal unistors; they cannot be treated as current sources controlled by their node voltage. However, they can be treated as normal unistors in topological formulas for network functions. Table I shows examples of some autonomic FUM's. These models were obtained on the basis of modified node equations of elements. It should be noted that the internal nodes of such models correspond to variables, which do not appear in normal node equations. Autonomic FUMs have two drawbacks: (i) they introduce additional vertices to the network graph, and (ii) some terms which can be cancelled may appear in symbolic network functions.

(i) and partially (ii) may be avoided with the use of nonautonomic models. Such models are obtained from modified SFG of elements, after removing internal vertices. The procedure leads to the necessity of changing the incidence of some graph edges in the neighbourhood of the element. In fact, nonautonomic model should be considered as algorithm of graph transformation rather than set of graph edges. Column 4 of Table I contains examples of formal-nullator-norator (FNN) SFGs of some elements. The FNN SFG describes algorithm mentioned. After all elements of the network are replaced by their SFG models (autonomic and/or FNN SFG), the following transformations have to be performed:

(i) transitors outgoing from one of the vertices of the nullator are to be moved to the second vertex the nullator is incident with (thus leaving the first vertex with zero outgoing degree).

(ii) transitors incoming to one vertex of the norator are to be moved to the other vertex of the norator (leaving the first one with zero incoming degree).

(iii) every vertex having zero outgoing degree should be identified with corresponding vertex of zero incoming degree. Note, the way the FNN SFGs are formed provides that this is always possible.

Replacing every transistor by three unistors (see Table I) we obtain the network unistor graph. This graph has fewer vertices and edges then graph with autonomic FUMs. The drawback of nonautonomic models lies in the fact, that SFG of the network has to be formed as intermediate result.

The procedure described shows the direct correspondence between network SFG and directed graph. Note, that formal unistors outgoing from circuit "ground" can be observed. This seems to contradict the unistor definition. However, the unistor graph origins from the IAM, and any vertex (not necessarily ground) can be chosen as reference node.

Matrix $\underset{\sim}{A}_a$, associated with network unistor graph G_u, is the matrix of the system of linear equations

$$\underset{\sim}{A}_a \; \underset{\sim}{x}_a = \underset{\sim}{F}_a \tag{7}$$

where

$$\underset{\sim}{x}_a = [x_1, \ldots, x_n]^T, \; \underset{\sim}{F}_a = [f_1, \ldots, f_n]^T$$

$$\underset{\sim}{A}_a = \begin{bmatrix} -\sum_{k=2}^{n} a_{k,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & -\sum_{\substack{k=1 \\ k\neq 2}}^{n} a_{k,2} & \cdots & a_{2,n} \\ \vdots & \vdots & & \vdots \\ a_{n,1} & a_{n,2} & \cdots & -\sum_{k=1}^{n-1} a_{k,n} \end{bmatrix} \tag{8}$$

System (7) is linearly dependent. To solve this system, we remove i-th equation and assume k-th node potential as reference. The determinant of $\underset{\sim}{A}_{ik}$, obtained from $\underset{\sim}{A}_a$ after removing i-th row and k-th column can be evaluated as [1].

$$\det \underset{\sim}{A}_{ik} = |T_k|$$

where T_k is the set of all directed trees with the k-th vertex as reference.

For simplicity, we assume i=k. Graph G_u may be extended taking into account excitations, and adding auxiliary edges. Fig. 3 shows how these edges can be added for the case of single excitation and single response. Multiple excitations and responses can be taken into account in a similar way.

The <u>extended graph</u> is associated with matrix

$$\underset{\sim}{A}_u = \left[\begin{array}{c|c} m & \underset{\sim}{L} \\ \hline -\underset{\sim}{F} & \underset{\sim}{A}_{kk} \end{array}\right] \tag{9}$$

where $\underset{\sim}{L} = [l_1, \ldots, l_{k-1}, l_{k+1}, \ldots, l_n]$

$\underset{\sim}{F} = [f_1, \ldots, f_{k-1}, f_{k+1}, \ldots, f_n]$.

It can be proved [7] that if $\underset{\sim}{A}_{kk}$ is nonsingular, then, for any $x_p \in \underset{\sim}{x}$,

$$x_p = \frac{\sum_{\substack{j=1 \\ j\neq k}}^{n} f_j \cdot |T_{(pj),(k)}|}{|T_k|} \tag{10}$$

where $T_{(pi),(k)}$ is the set of 2-trees t_V of G_u having the set of components $V = \{(p,j), (k)\}$.

To allow internal excitations, formal unistor models of independent sources can be introduced (Fig. 4).

Table II shows the set of formulas for basic network functions of a two-port network shown in Fig. 5 [1].

Table II

Topological Formulas

Function	Unistor graph	Chen digraph
$Z_i = \left.\frac{u_i}{i_i}\right\|_{i_o=0}$	$\frac{\lvert T_{(r),(s)}\rvert}{\lvert T_k\rvert}$ k-arbitrary	$\frac{\lvert T_{(r),(s)}\rvert}{\lvert T_k\rvert}$
$K_{uu} = \left.\frac{u_o}{u_i}\right\|_{i_o=0}$	$\frac{\lvert T_{(pr),(qs)}\rvert - \lvert T_{(qr),(ps)}\rvert}{\lvert T_{(r),(s)}\rvert}$	$\frac{\lvert T_{(rp),(sq)}\rvert - \lvert T_{(rq),(sp)}\rvert}{\lvert T_{(v),(s)}\rvert}$
$K_{ii} = \left.\frac{i_o}{i_i}\right\|_{u_o=0}$	$\frac{\lvert T_{(pr),(qs)}\rvert - \lvert T_{(qr),(ps)}\rvert}{\lvert T_{(p),(q)}\rvert}$	$\frac{\lvert T_{(rp),(sq)}\rvert - \lvert T_{(rq),(sp)}\rvert}{\lvert T_{(p),(q)}\rvert}$
$Z_{iu} = \left.\frac{u_o}{i_i}\right\|_{i_o=0}$	$\frac{\lvert T_{(pr),(qs)}\rvert - \lvert T_{(qr),(ps)}\rvert}{\lvert T_k\rvert}$ k-arbitrary	$\frac{\lvert T_{(rp),(sq)}\rvert - \lvert T_{(rq),(sp)}\rvert}{\lvert T_k\rvert}$
$Y_{iu} = \left.\frac{i_o}{u_i}\right\|_{u_o=0}$	$\frac{\lvert T_{(pr),(qs)}\rvert - \lvert T_{(qr),(ps)}\rvert}{D}$	$\frac{\lvert T_{(rp),(sq)}\rvert - \lvert T_{(rq),(sp)}\rvert}{D}$

where $D = \lvert T_{(rq),(s),(p)}\rvert + \lvert T_{(r),(sq)(p)}\rvert + \lvert T_{(rp),(s),(q)}\rvert + \lvert T_{(r),(sp),(q)}\rvert$

IV. GRAPH DECOMPOSITION

Unistor graph of electronic network should be hierarchically decomposed before run 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 worth to).

The decomposition can be made manually or automatically. A heuristic algorithm satisfying these conditions was proposed in [8] and implemented in a computer program described in section VII of the paper. The time of partition linearly depends on the number of graph nodes.

V. 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

$$T_V = \sum_{t \varepsilon Q_V} \prod_{G_i \varepsilon S} T^i_{V_t} \tag{11}$$

where Q_V is the set of all proper k-trees of the substitute graph of decomposition G^s. $T^i_{V_t}$ is the set of all k-trees of graph G_i having set of components V_t. V_t is determined as follows. Let $G^t_i \triangleq G^t_i(B_i, E^s_i \cap t)$, and let k be the number of components of G^t_i, where each of these components is of the form of incoming star, following the definition of proper k-tree. j-th component of G^t_i determines a component of V_t of the form $(r_i, v^j_1, \ldots, v^j_{m_j})$, where r_j is the reference node of that component, and v^j_l are remaining vertices of that component.

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

Remark:

(11) can be used recursively in the case any of graphs $G_i \varepsilon$ S was further partitioned.

Direct application of the Theorem 1 and Remark for hierarchically partitioned graphs leads to the algorithm of downward hierarchical analysis described in [5].

However, better results can be obtained by means of upward analysis described in what follows.

Definition. By the set P_B of multitrees of 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

$$B = \{r_1, v_1^1, \ldots, v_{m_1}^1, \ldots, r_k, v_1^k, \ldots, v_{m_k}^k\} \tag{12}$$

and summation is spreaded out over all possible V satisfying (12).

The following corollary follows immediately from Theorem 1 (cf. [10]).

Corollary:

$$P_B = \sum_{t \in R_B} \prod_{G_i \in S} T^i_{V_t} \tag{13}$$

where R_B is the set of proper multitrees of graph G^s spanned over B. $T^i_{V_t}$ is as described in Theorem 1.

Note that $T^i_{V_t}$ is a subset of P^i_B, where P^i_B is the set of all multitrees of G_i spanned over B_i. In conclusion, (13) allows us to determine the set P_B of graph G on the basis of the sets P^i_B of graphs $G^i \in S$ and on the structure of their interconnections G^s. P_B contains all information about graph considered as a n-pole network, where $n = \bar{B}$, and will be called a description of G. The Remark can be applied to (13). With above Corollary and remarks we can describe algorithm of upward analysis as follows.

Step 0. Decompose hierarchically the network graph; form the tree of decomposition. Find sets of block vertices for all proper and substitute graphs.

Step 1. Enumerate (directly) the set P_B^k of all multitrees of a proper block G_k spanned over block vertices B_k. Repeat this step for all proper blocks. Flag corresponding vertex in the tree of decomposition.

Step 2. Find such a vertex j in the tree of decomposition that all vertices originated from j (i.e. such vertices m_i, that there exists edge directed from j to m_i) are flagged. Use formula (13) to determine P_B^j and flag vertex j. Repeat step 2 until all vertices in the tree of decomposition are flagged.

Note, that descriptions of subgraphs of G has to be known before the description of G can be determined. The analysis moves "up" the tree of decomposition. This is opposite to the downward analysis, where analysis began at substitute graph of first partition and then moved "down" and "up" the tree of decomposition. Thus, upward analysis avoids multiple enumeration of the same sets of k-trees.

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 from Fig. 5 we have (compare Table II)

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

Because 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 subsets of P_B.

Similar formulas for other network functions can be easily derived on the basis of Table II.

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 time of analysis τ can be estimated as

$$\tau \simeq b.\tau_p + (b-1).\tau_s \simeq b(\tau_p + \tau_s) \qquad (14)$$

where τ_p and τ_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, (14) shows linear dependence on network size. Similar dependence for downward analysis was const.n^{α}, where n is number of network nodes and $\alpha \simeq 2 \div 3$ - cf. [5].

VI. ENUMERATION OF MULTITREES

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 simple 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. Of course, the sets of G_1 and G_2 are disjoint. The procedure can be applied recursively; it is finite because in each step we have graphs with fewer edges.

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

$$h_1, h_2, \ldots, h_\varepsilon, v_1, v_2, \ldots v_{n_B} \tag{15}$$

where $h_i \in V-B$, $i=1, \ldots, \varepsilon$, $v_i \in B$, $i=1, \ldots, n_B$

Definition. The structural matrix $M^{(1)} = [m_{ij}]$ of G is nxn matrix, where m_{ij} is the set of edges outgoing from vertex i and incoming to j; $m_{ii} \triangleq \emptyset$. Rows and columns of $M^{(1)}$ are enumerated according to (15).

Theorem 2

The set of all multitrees of G spanned over B can be obtained by iterative expansion of function $T(M^{(1)})$, with

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

$$T(M^{(n+1)}) = \{\emptyset\}$$

where $\{\emptyset\}$ denotes unity element of Wang algebra, $m_{ij} \in M^{(i)}$, $i = 1, \ldots, n$ and

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

$M_j^{(i+1)}$ is a matrix obtained from $M^{(i)}$ after adding i-th column of $M^{(i)}$ to j-th column, and setting i-th column to zero (for $i \neq j$; for $i = j$, $M_j^{(i+1)} = M^{(i)}$). Extracting m_{ii} denotes, that new component with reference node i appeared.

The set of multitrees has no duplications.

VII. COMPUTER PROGRAM

In this section we give some remarks on computer implementation of the algorithm of upward analysis.

The essential problem lies in data structures because of necessity of storing all sets of multitrees in the computer memory. Any multitree weight $|t|$ can be written as

$$|t| = C \cdot s^M \cdot \prod_{i=1}^{N} y_i^{Pi} \qquad (16)$$

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_V. Sets T_V will be called records of P_B. 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, ..., 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 sum of products with hierarchical parenthesis:

$$\sum_{Q_1} \prod_{S_1} \left(\ldots \left(\sum_{Q_k} \prod_{S_k} \left(\sum_{t \in T} \prod_{e \in t} y_e \right) \right) \ldots \right) \qquad (17)$$

Computing numerical values from (17) can be done without expanding the formula, because intermediate results are stored, thus saving

computer time. The form (17) 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 elaborated on the basis of presented algorithms. The program assumes the hierarchical bisection case only. This does not void loss of generality because any hierarchical decomposition can be presented as hierarchical bisection with the same proper blocks. The structure of the program is shown in Fig. 6.

The program makes use of nonautonomic unistor models. It accepts most of linear circuit elements. The network size which can be handled by the program is limited by the size of computer storage available. Program was tested for networks having up to 200 nodes. Fig. 7a 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 peformed for partitions resulting in proper blocks as shown in Fig. 7b. Fig. 7c 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.

VIII. EXAMPLES

To illustrate the method and show how it can be used in hand calculations, the circuit shown in Fig. 8 was chosen [9].

The network has been partitioned first to two subnetworks through the nodes 13-0, and each of these subnetworks was further partitioned as is shown in Fig. 8. The tree of decomposition is shown in Fig. 9. Fig. 10 shows substitute graphs of decomposition.

First, description of proper blocks ($G_1 \div G_7$) have to be determined. They are presented in Table III. We shall explain details on the example of blocks G_2, G_4, G_6. The unistor graph of these blocks is shown in Fig. 11. These blocks have the same structure and, therefore, we can analyse only one of them. Because this is 2-pole, we have 3 possible types of k-trees (records): trees $T_{(ab)}$ and $T_{(ba)}$ and 2-trees $T_{(a),(b)}$. Enumerating these sets we have

$$|T^i_{(ab)}| = |T^i_{(ba)}| = sC^i_1[(g_i + sC^i_2)(-g_i + g_i + sC^i_3) + (-g_i)(sC^i_3 - g_i)] =$$
$$= s^3 C^i_1 C^i_2 C^i_3 + s\, C^i_1 g^2_i$$

$$|T^i_{(a),(b)}| = sC^i_1(-g_i + g_i + s\, C^i_3) + (sC^i_2 - g_i)\cdot(-g_i + g_i + sC^i_3) +$$
$$g_i(sC^i_3 + g_i) = s^2 C^i_1 \cdot C^i_3 + s^2 C^i_2 \cdot C^i_3 + g^2_i \qquad i = 2, 4, 6$$

The description of graph G_8(union of graphs $G_1 \div G_4$) can be obtained from formula (13) and is presented in Table IV. For example, the first product of F_{15} can be obtained in the following way. The tree $t = \{a_1, a_4, b_1, d_1\}$ of the substitute graph of decomposition of G_8 is proper,

TABLE III

ANALYSIS OF PROPER BLOCKS

Graph	Type	Function	
G_1	$F_1=T^1_{(1)}=T^1_{(0)}=T^1_{(3)}$	$s^2C_1C_3g_1^2/R_1 + s^2C_2C_3g_1^2/R_1 + s^2C_1C_2g_1^2/R_1 + s^4C_1C_2C_3C_4/R_1$	
	$F_2=T^1_{(1,3),(0)}=T^1_{(3,1),(0)}$	$sC_2g_1^2/R_1 + s^3C_2C_3C_4/R_1$	
	$F_3=T^1_{(1,0),(3)}=T^1_{(0,1),(3)}$	$sC_1g_1^2/R_1 + s^3C_1C_3C_4/R_1$	
	$F_4=T^1_{(3,0),(1)}=T^1_{(0,3),(1)}$	$sC_3g_1^2/R_1 + s^2C_1C_2g_1^2 + s^2C_1C_3g_1^2 + s^2C_2C_3g_1^2 + s^4C_1C_2C_3C_4$	
	$F_5=T^1_{(0),(1),(3)}$	$g_1^2/R_1 + sC_1g_1^2 + sC_2g_1^2 + s^2C_3C_4/R_1 + s^3C_1C_3C_4 + s^3C_2C_3C_4$	
G_2 G_4 G_6	$F_{6,i}=T^i_{(a)}=T^i_{(b)}$	$sC_1^ig_i^2 + s^3C_1^iC_2^iC_3^i$	Remarks: i=2,4 or 6
	$F_{7,i}=T^i_{(a),(b)}$	$g_i^2 + s^2C_1^iC_3^i + s^2C_2^iC_3^i$	a = 3,8 or 13 b = 8,13 or 18
G_3	$F_{8,i}=T^i_{(a)}=T^i_{(0)}$	$sC_1^ig_i^2+sC_3^ig_i^2+s^3C_1^iC_2^iC_3^i$	a = 8 or 13
G_5	$F_{9,i}=T^i_{(a),(0)}$	$g_i^2 + s^2C_1^iC_2^i$	i = 3 or 5
	$F_{10}=T^7_{(18)}=T^7_{(21)}=T^7_{(0)}$	$sC_5g_7^2/R_2+sC_7g_7^2/R_2+s^2C_5C_7g_7^2+s^2C_5C_8g_7^2$ $+s^2C_7C_8g_7^2+s^3C_5C_6C_7/R_2+s^4C_5C_6C_7C_8$	

Table III (continued)

G_7	$F_{11}=T^7_{(18,21),(0)}=$ $T^7_{(21,18),(0)}$	$sC_7 \cdot g_7^2 + s^3C_5C_6C_7$
	$F_{12}=T^7_{(18,0),(21)}=$ $T^7_{(0,18),(21)}$	$sC_5g_7^2$
	$F_{13}=T^7_{(18),(21,0)}=$ $T^7_{(18),(0,21)}$	$g_7^2/R_2 + sC_8 \cdot g_7^2 + s^2C_5C_6/R_2 + s^3C_5C_6C_8$
	$F_{14}=T_{(18),(0),(21)}$	$g_7^2 + s^2C_5C_6$

and $t \in T^8_{(1,13,0)}$. The intersections of t and substitute graphs representing successive proper blocks $G_1 \div G_4$ are: $E_1^s \cap t = \{a_1, a_4\}$, $E_2^s \cap t = \{b_1\}$, $E_3^s \cap t = \{\emptyset\}$, $E_4^s \cap t = \{d_1\}$. Thus, we have following sets of components for graphs G_1, G_2, G_3 and G_4, respectively: $V_1^t = \{(1,3,0)\}$, $V_2^t = \{(3,8)\}$, $V_3^t = \{(8),(0)\}$, $V_4^t = \{(8,13)\}$ (as there is no edges of t in G_3^s, all block vertices of G_3 are isolated, and thus $V_3^t = \{(8),(0)\}$).

Remaining components in Table IV are obtained in the same way. The procedure can be repeated to obtain the description of graph G_9, (union of graphs $G_5 \div G_7$), and then the description of graph $G_{10} = G$ (union of graphs G_8 and G_9). Having the description of G, any network function can be evaluated, for example:

$$k_u = \frac{F_{26}}{F_{26} + F_{28}} \quad , \quad Z_{in} = \frac{F_{26} + F_{28}}{F_{25}} \quad \text{etc.}$$

IX. CONCLUSIONS

This paper presents a method for topological analysis of large linear networks which are represented by unistor graphs.

TABLE IV

ANALYSIS OF SUBSTITUTE GRAPHS

Graph	Type	Function
G_8	$F_{15}=T^8(1)=T^8(13)=T^8(0)$	$F_1 \cdot F_{6,2} \cdot F_{9,3} \cdot F_{6,4} + F_2 \cdot F_{6,2} \cdot F_{8,3} \cdot F_{6,4} + F_1 \cdot F_{7,2} \cdot F_{8,3} \cdot F_{6,4} + F_3 \cdot F_{6,2} \cdot F_{8,3} \cdot F_{6,4}$
	$F_{16}=T^8(1,13),(0)=T^8(13,1),(0)$	$F_2 \cdot F_{6,2} \cdot F_{9,3} \cdot F_{6,4}$
	$F_{17}=T^8(1,0),(13)=T^8(0,1),(13)$	$F_3 \cdot F_{6,2} \cdot F_{9,3} \cdot F_{6,4} + F_1 \cdot F_{7,2} \cdot F_{9,3} \cdot F_{6,4} + F_1 \cdot F_{6,2} \cdot F_{9,3} \cdot F_{7,4} + F_1 \cdot F_{7,2} \cdot F_{8,3} \cdot F_{7,4} + F_3 \cdot F_{6,2} \cdot F_{8,3} \cdot F_{7,4} + F_2 \cdot F_{6,2} \cdot F_{8,3} \cdot F_{7,4}$
	$F_{18}=T^8(1),(13,0)=T^8(1),(0,13)$	$F_2 \cdot F_{7,2} \cdot F_{8,3} \cdot F_{6,4} + F_4 \cdot F_{7,2} \cdot F_{8,3} \cdot F_{6,4} + F_5 \cdot F_{6,2} \cdot F_{8,3} \cdot F_{6,4} + F_4 \cdot F_{6,2} \cdot F_{9,3} \cdot F_{6,4}$
	$F_{19}=T^8(1),(0),(13)$	$F_2 \cdot F_{6,2} \cdot F_{9,3} \cdot F_{6,4} + F_4 \cdot F_{7,2} \cdot F_{8,3} \cdot F_{7,4} + F_5 \cdot F_{6,2} \cdot F_{9,3} \cdot F_{6,4} + F_2 \cdot F_{7,2} \cdot F_{9,3} \cdot F_{6,4} + F_2 \cdot F_{7,2} \cdot F_{8,3} \cdot F_{7,4} + F_4 \cdot F_{7,2} \cdot F_{9,3} \cdot F_{6,4}$
G_9	$F_{20}=T^9(13)=T^9(0)=T^9(21)$	$F_{9,5} \cdot F_{6,6} \cdot F_{10} + F_{8,5} \cdot F_{7,6} \cdot F_{10} + F_{8,5} \cdot F_{6,6} \cdot F_{11} + F_{8,5} \cdot F_{6,6} \cdot F_{13}$
	$F_{21}=T^9(13,21),(0)=T^9(21,13),(0)$	$F_{9,5} \cdot F_{6,6} \cdot F_{11}$
	$F_{22}=T^9(13,0),(21)=T^9(0,13),(21)$	$F_{9,5} \cdot F_{6,6} \cdot F_{14} + F_{8,5} \cdot F_{7,6} \cdot F_{12} + F_{8,5} \cdot F_{7,6} \cdot F_{11} + F_{9,5} \cdot F_{6,6} \cdot F_{12}$

Table IV cont'd

	$F_{23}=T^{9}_{(21,0),(13)}=T^{9}_{(0,21),(13)}$	$F_{9,5}\cdot F_{6,6}\cdot F_{13} + F_{9,5}\cdot F_{7,6}\cdot F_{10}$
	$F_{24}=T^{9}_{(21),(0),(13)}$	$F_{9,5}\cdot F_{6,6}\cdot F_{14} + F_{9,5}\cdot F_{7,6}\cdot F_{12} + F_{9,5}\cdot F_{7,6}\cdot F_{11}$
G_{10}	$F_{25}=T^{10}_{(1)}=T^{10}_{(21)}=T^{10}_{(0)}$	$F_{16}\cdot F_{20} + F_{17}\cdot F_{20} + F_{15}\cdot F_{21} + F_{15}\cdot F_{23}$
	$F_{26}=T^{10}_{(1,21),(0)}=T^{10}_{(21,1),(0)}$	$F_{16}\cdot F_{21}$
	$F_{27}=T^{10}_{(1,0),(21)}=T^{10}_{(0,1),(21)}$	$F_{15}\cdot F_{24} + F_{17}\cdot F_{22} + F_{17}\cdot F_{21} + F_{16}\cdot F_{22}$
	$F_{28}=T^{10}_{(1),(21,0)}=T^{10}_{(1),(0,21)}$	$F_{19}\cdot F_{21} + F_{18}\cdot F_{22} + F_{16}\cdot F_{23} + F_{18}\cdot F_{21}$
	$F_{29}=T^{10}_{(1),(21),(0)}$	$F_{16}\cdot F_{24} + F_{18}\cdot F_{24} + F_{19}\cdot F_{22} + F_{19}\cdot F_{21}$

The full symbolic analysis of large networks is efficiently realized by this method. It was proved that 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. North-Holland P.C., 1976.

[2] W.K. Chen, "Unified theory on the generation of trees of a graph", Int. J. Electr., "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. Elektrot., z.2, 1975, (in Polish), pp. 245-262.

[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", International Journal of Circuit Theory and Applications, vol. 8, 1980, pp.407-417.

[6] G. Centkowski, J. Starzyk and E. Sliwa, "Computer implementation of topological method in the analysis of large networks", Proc. European Conf. Circuit Theory and Design, Warsaw, 1980, pp. 69-74.

[7] J. Starzyk, Topological Analysis of Large Electronic Circuits. Praca habilitacyjna, Technical University of Warsaw, 1981 (in Polish).

[8] J. Starzyk, "An efficient cluster algorithm", Proc. of 5th Czech-Polish Workshop on C.T., 1980.

[9] D.F. Sheahan and H.J. Orchard, "Bandpass-filter realisation using gyrators", Electronic Letters, vol. 3, 1967, pp. 40-41.

[10] G. Centkowski, J. Starzyk and E. Sliwa, "Symbolic analysis of large LLS networks by means of upward hierarchical analysis", Proc. European Conf. Circuit Theory and Design, The Hague, 1981, pp. 358-361.

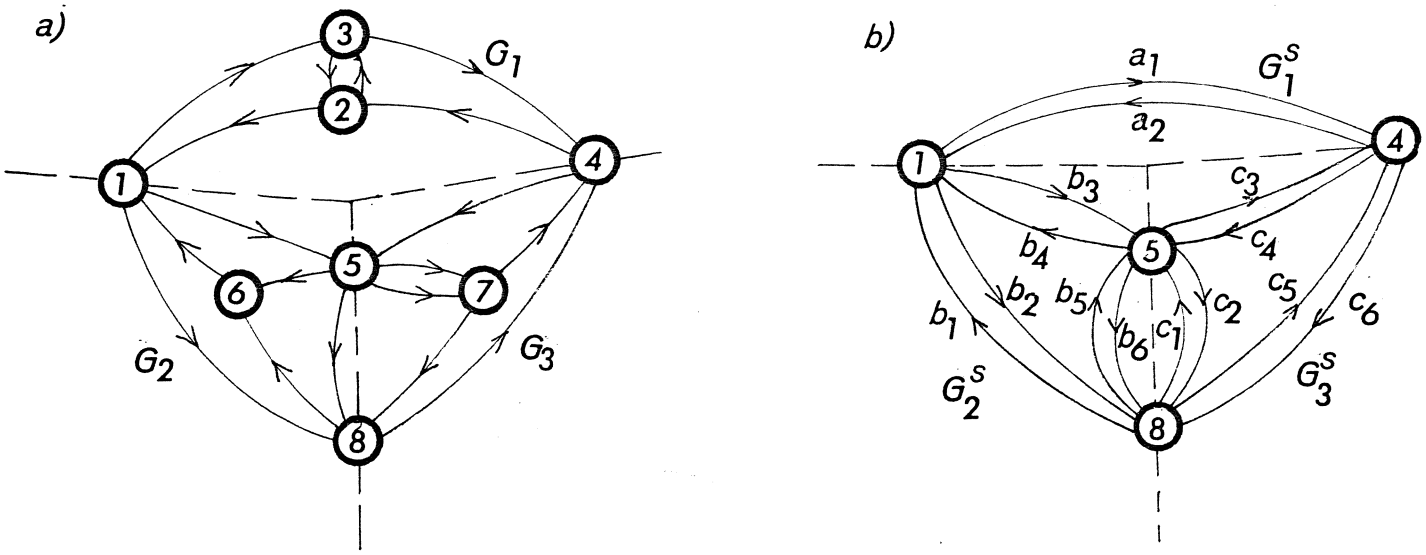


Fig. 1. Graph G and its trisection (a), and substitute graph of decomposition G^s (b).

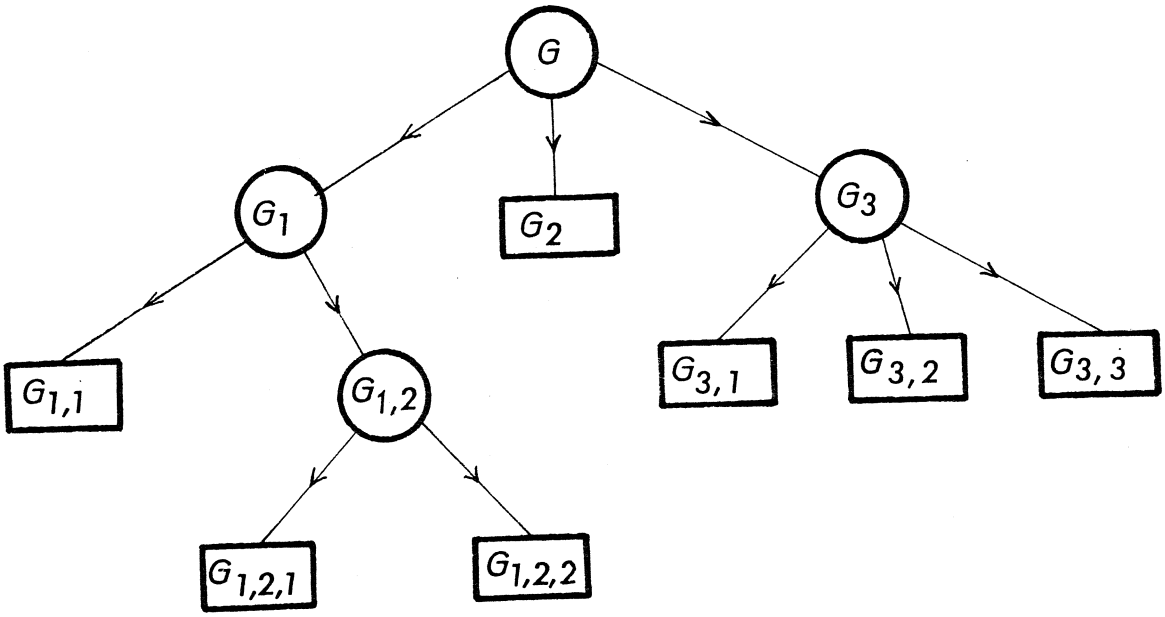


Fig. 2. Example of the tree of decomposition.

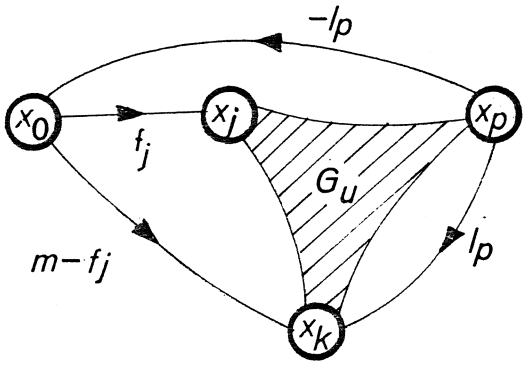


Fig. 3. Network with excitations and auxiliary edges.

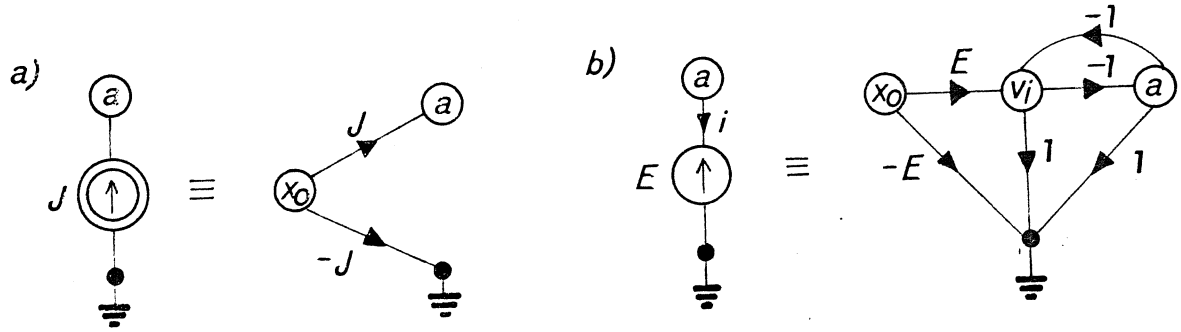


Fig. 4. FUMs of current (a), and voltage (b) independent sources.

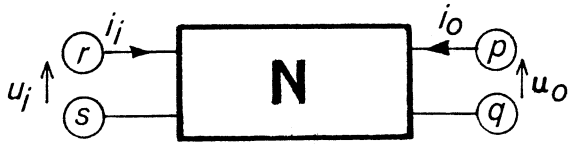


Fig. 5. Two-port network.

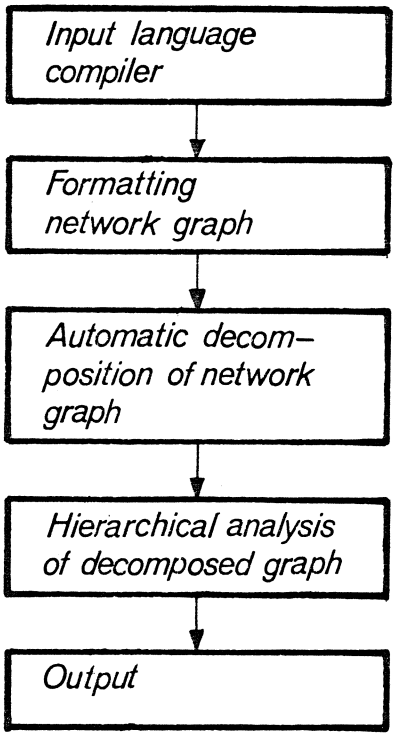


Fig. 6. Structure of HADEN2.

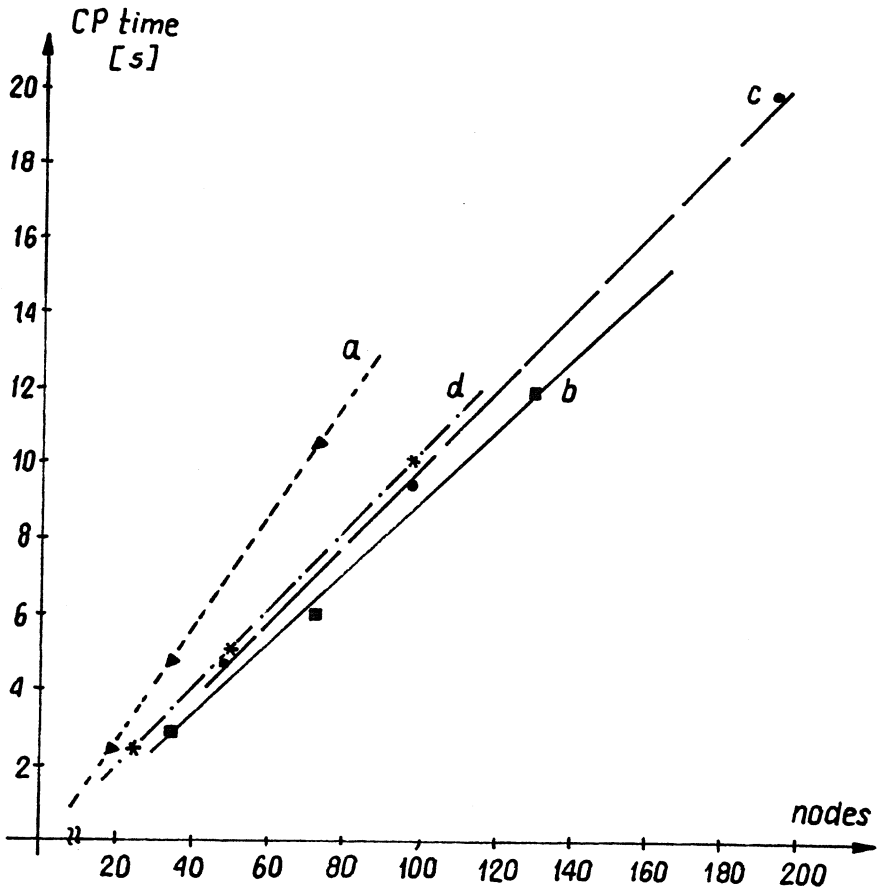


Fig. 7(a). Dependency of time of analysis w.r.t. the number of network nodes.

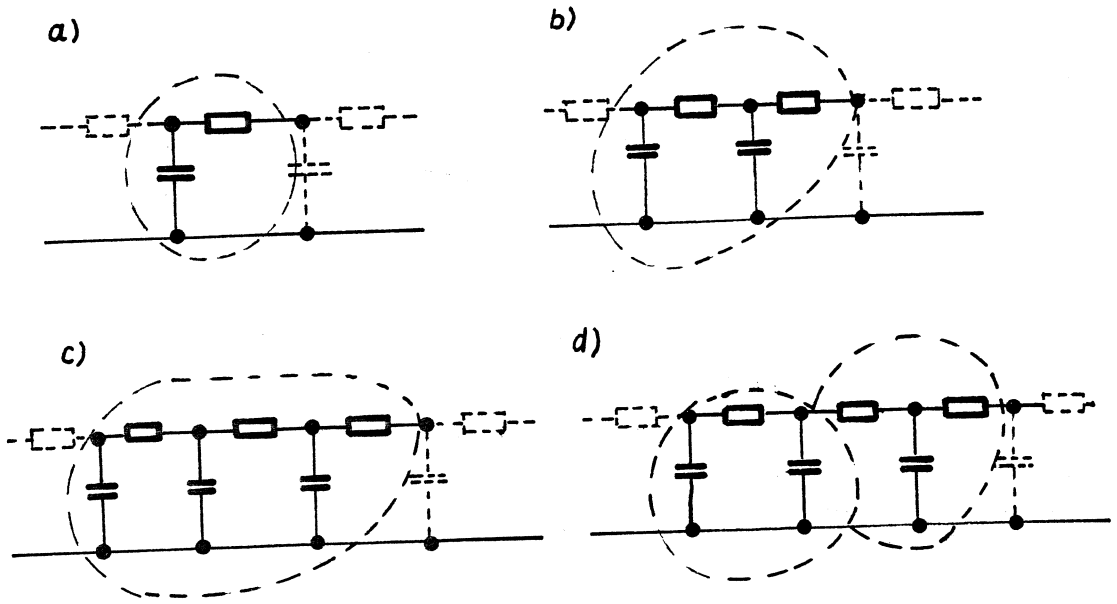


Fig. 7(b). Proper blocks of ladder decomposition.

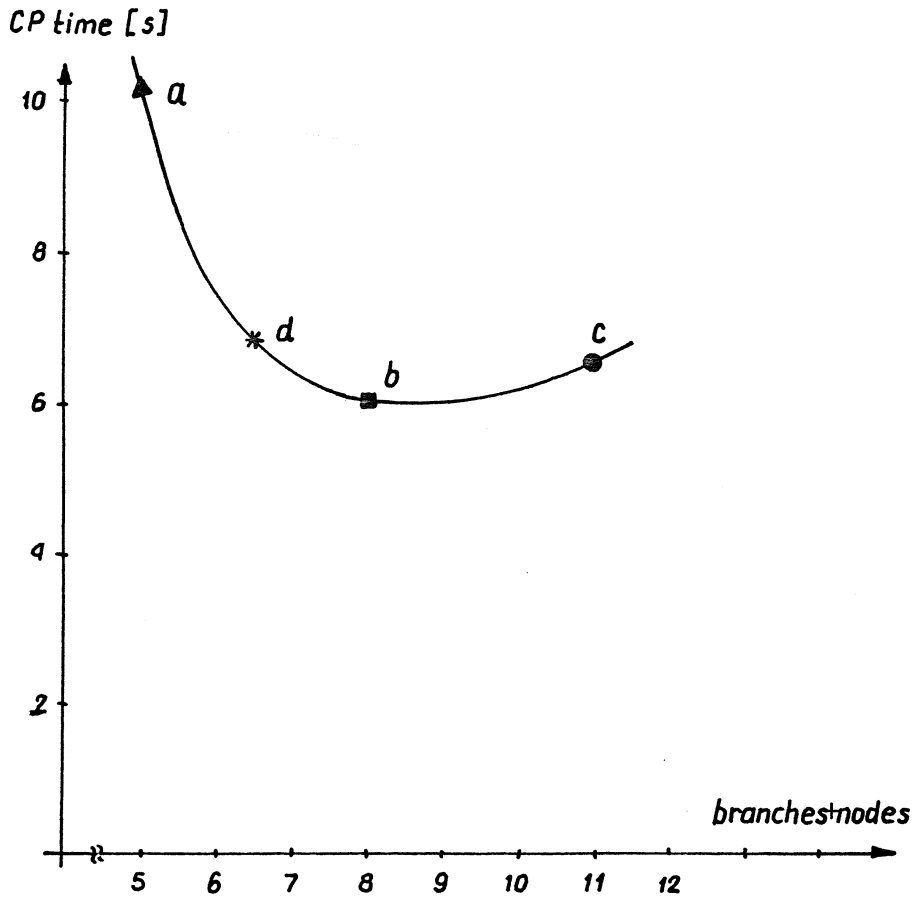


Fig. 7(c). Dependency of time of analysis w.r.t. the size of proper blocks.

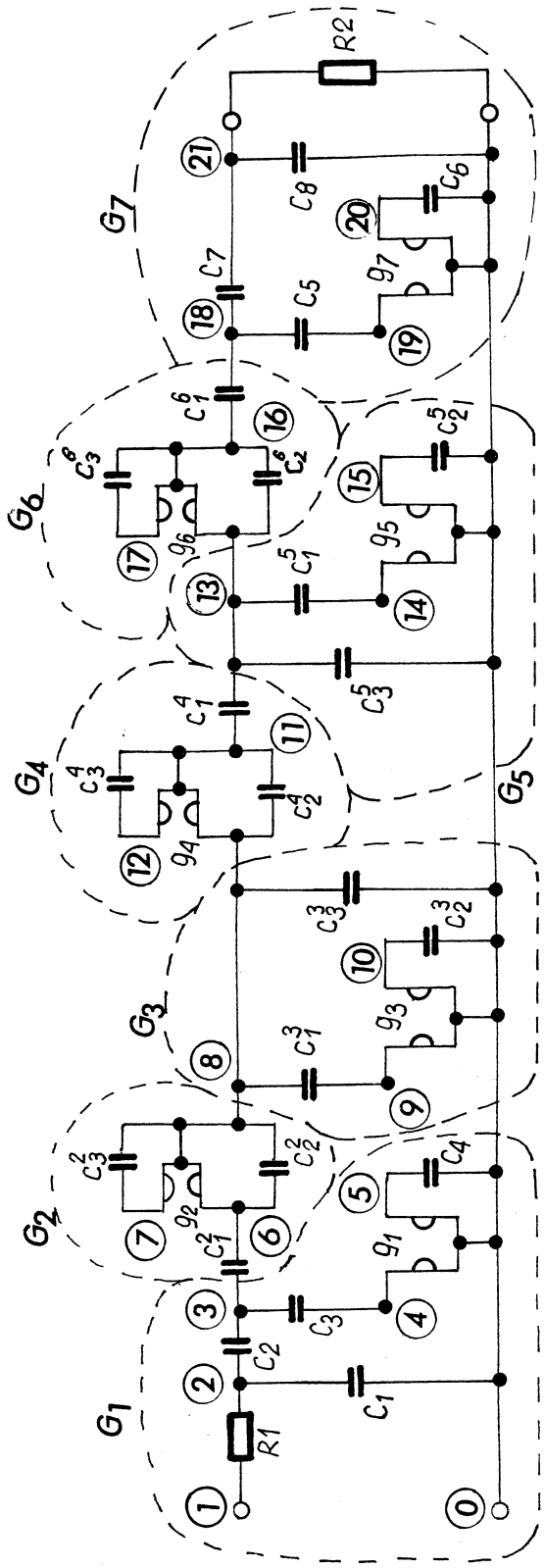


Fig. 8. Passive network and its decomposition.

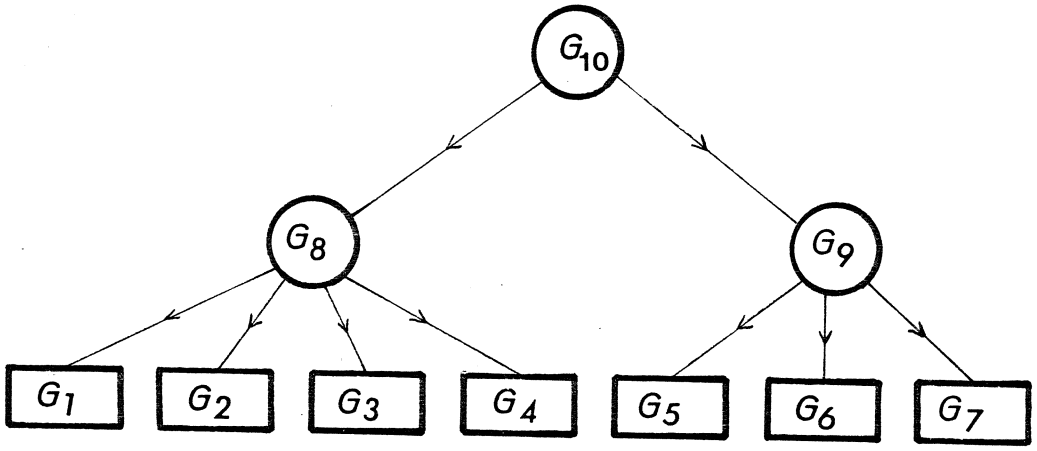


Fig. 9. The tree of decomposition of network from Fig. 8.

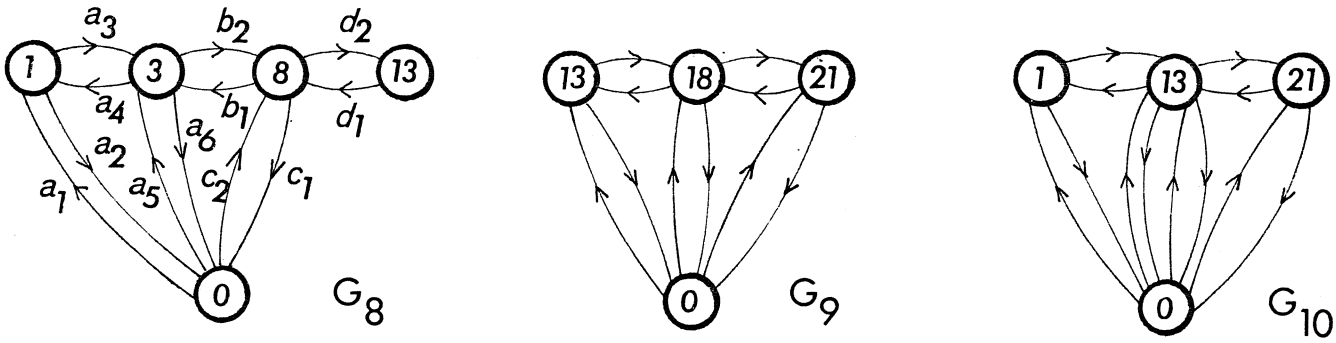


Fig. 10. Substitute graphs of succesive decompositions.

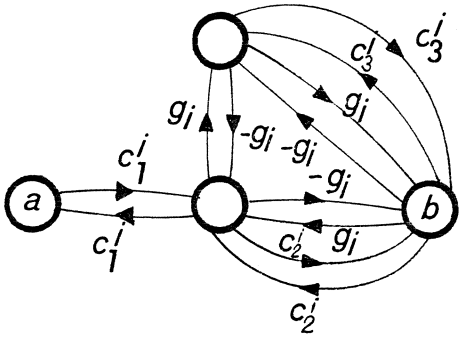


Fig. 11. Unistor graph of blocks G_2,G_4,and G_6.

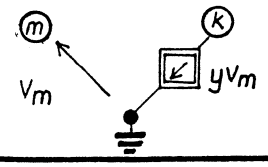
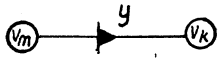
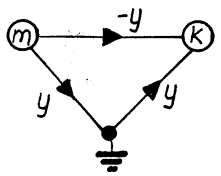
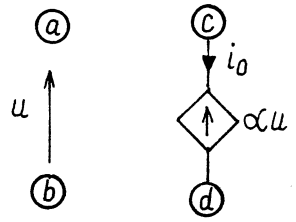
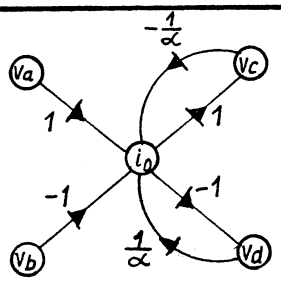
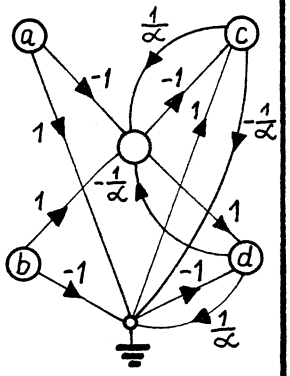
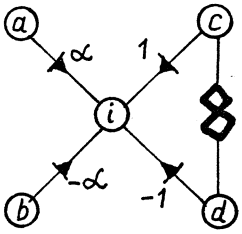
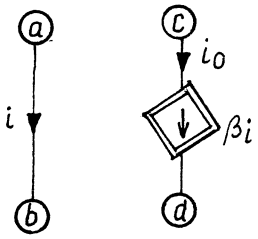
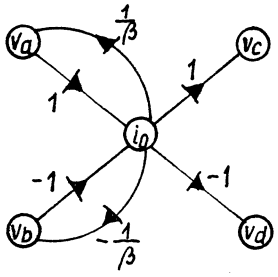
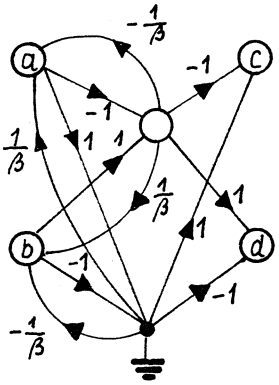
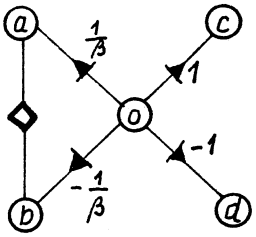
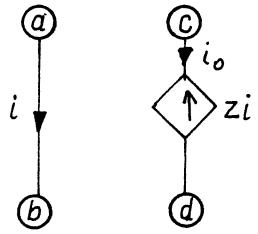
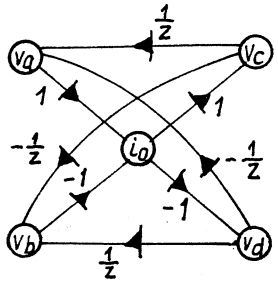
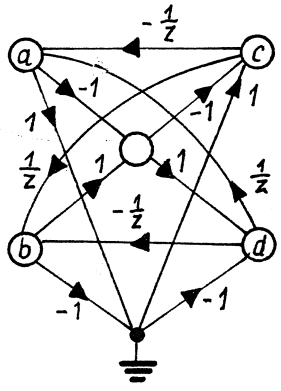
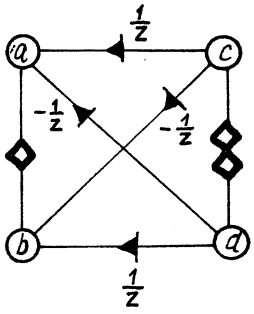
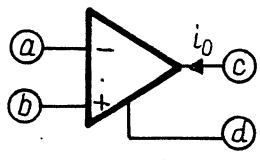
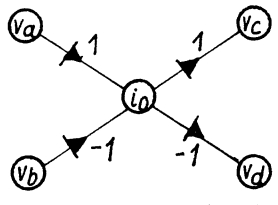
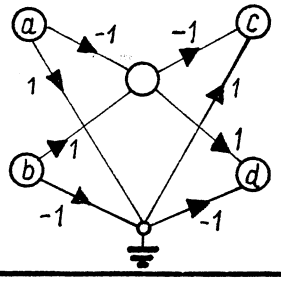
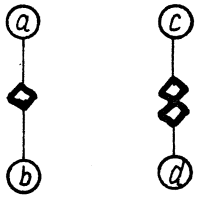
Element	SFG	Autonomic FUM	FNN flowgraph
Transitor 			
VVT 	 $V_a - V_b - \frac{1}{\alpha}(V_c - V_d) = 0$ $i_a = i_b = 0$ $i_c = -i_d = i_0$		
CCT 	 $V_a - V_b = 0$ $i_a = -i_b = \frac{1}{\beta} i_0$ $i_c = -i_d = i_0$		
CVT 	 $V_a - V_b = 0$ $i_a = -i_b = \frac{1}{z}(V_c - V_d)$ $i_c = -i_d = i_0$		
Opamp 	 $i_a = i_b = 0 \quad i_c = -i_d = i_0$		

TABLE I. UNISTOR MODELS OF NETWORK ELEMENTS.

SOC-273

UPWARD TOPOLOGICAL ANALYSIS OF LARGE CIRCUITS USING DIRECTED GRAPH REPRESENTATION

J.A. Staryzk and E. Sliwa

November 1981, No. of Pages: 33

Revised:

Key Words: Topological methods, linear network analysis, unistor graph, symbolic analysis of large networks, computer aided techniques

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. A new approach, using Coates signal-flow graphs, to the element modelling is described.

An algorithm of upward hierarchical analysis of 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.

Description:

Related Work:

Price: $ 6.00.