INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-306

FLOWGRAPH ANALYSIS OF LARGE ELECTRONIC NETWORKS

J.A. Starzyk and A. Konczykowska

January 1983

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

FLOWGRAPH ANALYSIS OF LARGE ELECTRONIC NETWORKS

J.A. Starzyk and A. Konczykowska

Abstract

The paper presents a new complete method for signal flowgraph analysis of large electronic networks. Two efficient methods of flowgraph formation that can easily represent decomposed networks are introduced. Hierarchical decomposition approach is realized using the so-called upward analysis of decomposed network. This approach removes the limitations on topological analysis and allows to obtain fully symbolic network formulas in time which is linearly proportional to the size of the network. The approach can be used to obtain symbolic solution of any linear system of equations.

A. Konczykowska and J.A. Starzyk are 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.

A. Konczykowska is with the Centre National d'Etudes des Telecommunication, 92220 Bagneux, France.

I. INTRODUCTION

The notion of topological analysis of electrical networks is connected with the determination of the network characteristics from the knowledge of elements and their connections (network topology) without applying the numerical methods of solving system of equations. As a result, for linear, lumped, stationary (LLS) networks, we get the transfer functions defined as the ratio of the Laplace transform of the input and output signals. The result can be presented in one of the following forms:

- in all-symbolic form, as a function of all network parameters;
- in semi-symbolic form, when only some parameters are kept as symbols;
- in a rational function form of complex frequency s;
- in tabulated form of values of transfer function for given frequency points.

Recently, the important problem of analyzing large networks was discussed in many monographs and articles [8,11,24,25,48,69,70,71,90,96, 101,126]. The main aim of these papers was to improve the computational efficiency (i.e. accuracy and time of computation, memory consumption, convergence of the algorithms, etc.) and gaining the full qualitative information about the network. The latter goal could be achieved by such a method of analysis that provides the results in all- or semi-symbolic form [2,41,73,102,103,106,107].

As the topological methods met the difficulties in analysing large networks, the other numerical approaches have been extensively elaborated and utilized such as: numerical methods for solving system of equations [10,24,46,49] and numerical methods for solving the eigenvalues problem [18,55,72,76,91,99,108].

This paper is consecrated to the problems of topological analysis of large linear networks. We only consider the Coates flowgraph representation of the network. Similar approach is possible with other representations (e.g. unistor graph [30]).

II. GENERAL CHARACTERIZATION OF TOPOLOGICAL METHODS

Methods of topological analysis started in the nineteenth century with the Kirchhoff's works [60]. These methods have been intensively developed in the sixties [9,16,21,30,35,39,54,89,117] when their expansion was connected with the popularization of the computer techniques. At the same time, the methods of abstract algebra have been elaborated. These methods allow the algebraic notation of the network topology and formalization of topological operations. It is worthy to mention the interest in Wang algebra [37,119,120,124] and Bellert algebra of structural numbers [3,4,5,6,42].

In that period one of the most important advantages of topological methods was the possibility of getting the transfer function directly from the network model [104] or the signal flow graph [29]. It was possible to study the direct impact of any network element on the results [29,77]. The efficiency of computation was considered as secondary because of the quick development of the computer techniques.

A special feature of topological methods, independently on the utilised graph representation, is the possibility of determination of network transfer function in a form of rational function. Numerator and denominator of this function are expressed as sum of products of graph edges weights [19]:

$$K(s) = \frac{L(s)}{M(s)} = \frac{\sum_{l} \prod_{i} y_i}{\sum_{m} \prod_{j} y_j} . \qquad (1)$$

These weights depend directly on the type and value of network elements.

Equation (1) is obtained from the solution of a set of linear equations describing the analysed network:

$$\underset{\sim}{A}\underset{\sim}{X} = \underset{\sim}{F} . \qquad (2)$$

It is well known from the linear algebra that the knowledge of determinant and cofactors of $\underset{\sim}{A}$ is sufficient to determine the solution $\underset{\sim}{X}$ of the system of equations (2).

Different topological representations take advantage of the relationship between the network topology and the coefficient matrix $\underset{\sim}{A}$.

For example if $\underset{\sim}{A}$ is the <u>nodal admittance matrix</u>, then

$$\underset{\sim}{A} = \underset{\sim}{\lambda}_1 \underset{\sim}{Y}_b \underset{\sim}{\lambda}_2^T , \qquad (3)$$

where

$\underset{\sim}{Y}_b$ - diagonal matrix of element admittances

$\underset{\sim}{\lambda}_1$, $\underset{\sim}{\lambda}_2$ - reduced incidence matrices [36] of graphs representing the network or their modifications obtained when some elements are put to zero.

Matrices $\underset{\sim}{\lambda}_1$ and $\underset{\sim}{\lambda}_2$ have the same number of rows and columns.

With different topological representations matrices $\underset{\sim}{\lambda}_1$, $\underset{\sim}{\lambda}_2$ have the following meaning:

1. Incidence matrices for current and voltage graphs [104] or nullator and norator graphs [32,33] (pair of conjugated graphs). In the particular case of RLC networks, $\underset{\sim}{\lambda}_1 = \underset{\sim}{\lambda}_2$.

2. Unistor [30] or dispersor graph [113] (directed graph). In the case of the unistor graph, $\underset{\sim}{\lambda}_1$ is the incidence matrix and $\underset{\sim}{\lambda}_2$ is obtained by setting to zero all -1 elements in $\underset{\sim}{\lambda}_1$. For the dispersor graph $\underset{\sim}{\lambda}_2$ is the incidence matrix of the graph and $\underset{\sim}{\lambda}_1$ is obtained by setting to zero all -1 elements of $\underset{\sim}{\lambda}_2$.

3. Coates graph [29] or Mason graph [77] (signal flowgraph). $\underset{\sim}{\lambda}_1$ is obtained from the incidence matrix by setting all +1 elements to 0 and changing the signs of other elements. $\underset{\sim}{\lambda}_2$ is obtained from the incidence matrix by setting to zero all -1 elements. For Mason graph equation (2) is replaced by $\underset{\sim}{A}^M \underset{\sim}{X} - \underset{\sim}{F} = \underset{\sim}{X}$ and $\underset{\sim}{A}^M = \underset{\sim}{A} + \underset{\sim}{I}$.

The relationship between the network and its topological representation is generally established by determining topological models of the network elements. This allows the determination of the graph directly from the network without the formulation of system of equations. Topological models are normally formed on the basis of the nodal admittance matrix (or indefinite admittance matrix [26]). Such approach gives direct representation for two-terminal and VCCS elements. Representations of other elements may be derived by combining the above two elements. However, this leads sometimes towards too much complicated models. For a pair of conjugated graphs, it is possible to get independent models for all basic elements of linear active networks [7,13,32,34,112].

Topological formulas which permit the computation of the determinant and cofactors of matrix $\underset{\sim}{A}$ of (2) are derived from the Binet-Cauchy theorem [36] applied to (3). In this theorem, if $\underset{\sim}{Q}$ and $\underset{\sim}{R}$ are matrices with the dimensions kxm and mxk respectively and $k \leq m$ then

$$\det (\underset{\sim}{Q}\underset{\sim}{R}) = \sum_{i_1 i_2 \ldots i_k} \det \underset{\sim}{Q}_{i_1 i_2 \ldots i_k} \det \underset{\sim}{R}^{i_1 i_2 \ldots i_k} , \qquad (4)$$

where

$\underset{\sim}{Q}_{i_1, \ldots i_k}$ - submatrix of Q formed of columns $i_1, \ldots i_k$,

$\underset{\sim}{R}^{i_1, \ldots i_k}$ - submatrix of R formed of rows $i_1, \ldots i_k$,

$i_1 i_2, \ldots i_k$ - combination of k elements out of m.

It can also be shown that the determinant of main submatrices of incidence matrix differs from 0 (and equals +1 or -1) if and only if we deal with the incidence matrix of a tree [104].

In accordance to the Binet-Cauchy theorem the determinant of matrix $\underset{\sim}{A}$ of (3) for the chosen representation is equal to:

1. Algebraic sum of <u>complete trees weights</u> [63,88,104] for current and voltage graphs (or nullator and norator graphs) when the representation by pair of conjugated graphs is used. A weight of a complete tree is equal to a product of all tree edge weights, and the <u>tree sign</u> is equal to the product of corresponding main determinants of $\underset{\sim}{\lambda}_1$ and $\underset{\sim}{\lambda}_2$.
2. Sum of <u>directed trees weights</u> [19] in the case of directed graph representation. These trees are directed towards a chosen reference node. In the unistor graph it is a consequence of nonsingularity of a submatrix of $\underset{\sim}{\lambda}_2$, corresponding to a set of edges such that from each

node except the reference there is exactly one starting edge. In the dispersor graph we have the similar nonsingularity of submatrix of λ_1. A weight of a tree is equal to the product of all tree edge weights.

3. Algebraic sum of connection weights [29] when the representation by signal flow-graphs is considered. In the Coates graph a connection is a set of disjoint loops incident with all graph nodes. A weight is defined as the product of all edge weights and a sign is equal to $(-1)^{l_p}$, where l_p is the number of loops in the connection.

Matrices λ_1 and λ_2 are formed from the incidence matrix by setting to zero respectively 1 and -1 elements. They have at most one non-zero element in each column. It is then obvious that maximal nonsingular submatrices of λ_1 and λ_2, will have exactly one non-zero element in each row and column. A pair of such submatrices correspond to a subgraph in which each node is incident exactly with one incoming and one outcoming edge. It is equivalent to the statement that this subgraph is a connection of a flowgraph. This argumentation can be considered as proof of Mason rule [77,78] for a determinant of Coates graph. A proof of Mason rule that is recognized as a particularly elegant can be found in a paper by Kim and Chien [59].

Realization of topological formulas requires the knowledge of all graph trees or connections. To make the computations efficient one should use the algorithms which generate trees or connections rapidly and without duplications. Only in this case, it is possible not to check any new tree or connection with all previously generated ones. There are many efficient algorithms for graph trees [17,23,47,81,83,86, 95] and connections [98,118,124] generation. They form the basis of many programs of topological analysis intended for the analysis of small linear networks [28,29,74,84,88].

In practice, direct application of topological formulas permits the analysis of networks with graph having about 10 nodes [26]. This limitation is not the result of the small efficiency of generation algorithms but of the great number of terms in the topological formulas for the determinant of the coefficient matrix in (3). Even if we could generate all terms in null time, the time needed for weights evaluation would grow proportionally (or quicker) to the number of terms and for relatively small networks (with about 20 nodes) will attain enormous values.

So the direct realization of topological formulas is practically impossible for networks having more than 20 nodes, even for the fastest computers. The foresighted progress in the computer techniques can only slightly move this "limit of physical realizability" of direct topological formulas. Besides it is obvious that application of topological formulas for networks having more than 10 nodes is much more time consuming then the methods of symbolic analysis based on the numerical techniques of determinant evaluation [2,12,41,121].

These are the reasons why the methods of topological analysis based on trees and connections generation were judged by McCall and Pederson [82] as inefficient. Nevertheless, the investigations in the domain of topological analysis and design have been carried on [1,51,56,58,61,62, 85,87,92,94,97].

Trials of introducing methods of graph reduction [20,31,38] or decomposition [22,44,105] to the analysis didn't provide the universal efficient programs of analysis and were poorly estimated in paper by Alderson and Lin [2].

The investigations on the possibility of application of decomposition techniques have been continued [123]. In the paper by Chen [22], the problem of trees generation has been solved for the case of n-node bisection and in Konczykowska and Starzyk [68] for the general case of simple decomposition. These achievements were the basis of the method and the program of topological analysis of networks represented by a pair of conjugated graphs [63,64].

Computer realization by simple decomposition method allowed an increase by 2-4 times the size of analyzed networks. But still the topological analysis of large networks having more than 100 nodes was unreal.

Besides, the realized method for simple decomposition had many limitations on the graph partitioning, and computations were considerably complicated in the case of decomposition with defect [63].

An important progress in the domain of topological analysis has been achieved with the introduction of hierarchical decomposition. In the paper by Starzyk [111] the method of signal flowgraph analysis has been presented and in Starzyk and Sliwa [110,114] the hierarchical analysis of directed graph has been discussed. Based on these methods the programs of topological analysis of large networks have been elaborated [15,65,67].

Further improvement has been attained when the upward hierarchical method was introduced [14,66]. The details of this method will be presented in the paper. The main aim of the investigations is to reduce the time consumption from the involution dependence for the previous (downward) decomposition to the linear dependency.

III. SIGNAL FLOWGRAPH OF A NETWORK

There are many ways of forming a flowgraph for an electrical network. When the computer techniques are applied, the primitive [79] or compact [26] Mason graph is the most frequently utilised. A primitive graph illustrates the set of element, KCL and KVL equations. Graph nodes represent all element voltages and currents. Compact graph can be obtained from the primitive graph by elimination of nodes corresponding to the currents in the branches of chosen tree and voltages of the branches of corresponding cotree [36]. The theory of matrix Coates graphs [27] has been elaborated to obtain topological representation of matrix operations. These graphs are specified by matrix equations (2) with the submatrices as coefficient matrix elements, which leads to the definition of a graph representing the matrix system of equations

$$\begin{bmatrix} \underset{\sim}{A}_{11} & \underset{\sim}{A}_{12} & \cdots & \underset{\sim}{A}_{1n} \\ \underset{\sim}{A}_{21} & \underset{\sim}{A}_{22} & \cdots & \underset{\sim}{A}_{2n} \\ \vdots & & & \\ \underset{\sim}{A}_{n1} & \underset{\sim}{A}_{n2} & \cdots & \underset{\sim}{A}_{nn} \end{bmatrix} \begin{bmatrix} \underset{\sim}{X}_1 \\ \underset{\sim}{X}_2 \\ \vdots \\ \underset{\sim}{X}_n \end{bmatrix} - \begin{bmatrix} \underset{\sim}{F}_1 \\ \underset{\sim}{F}_2 \\ \vdots \\ \underset{\sim}{F}_n \end{bmatrix} = \underset{\sim}{0} \ . \tag{5}$$

In a similar way, it is possible to define the Mason matrix graph which represents the following equations:

$$\underset{\sim}{A}^M \underset{\sim}{X} - \underset{\sim}{F} = \underset{\sim}{X} \tag{6}$$

with submatrices as matrix elements, and $\underset{\sim}{A}^M = \underset{\sim}{A} + \underset{\sim}{I}$. The nodes of Mason or Coates matrix graph correspond to the subsets of variables (voltages or currents) and submatrices $\underset{\sim}{A}_{ij}$ (i,j = 1,2,..n) are the weights of edges. In Fig. 1 primitive and compact Mason matrix graphs are represented.

Symbols in the figure have the following meaning:

- $\underset{\sim}{E}$ - vector of independent voltage sources, they must be tree branches
- $\underset{\sim}{J}$ - vector of independent current sources, they must be cotree branches
- $\underset{\sim}{U}_T, \underset{\sim}{I}_T$ - vectors of voltages and currents in remaining tree branches
- $\underset{\sim}{U}_L, \underset{\sim}{I}_L$ - vectors of voltages and currents in remaining cotree branches
- $\underset{\sim}{Z}_T, \underset{\sim}{Y}_L$ - matrices of element equations (for RLC networks these matrices are diagonal).

Evidently a structure of a flowgraph depends on the chosen tree and this leads to various possible flowgraphs for any electrical network. When choosing the tree in order to form a flowgraph, we should try to obtain a graph with the simplest structure and the minimal number of edges. It is not always possible to be satisfied with the graph obtained on the basis of chosen tree.

An inconvenience for both primitive and compact types of graphs is the necessity of choosing the optimal tree, and the structure of the resulting graph is difficult to anticipate. Moreover, it is not possible to create the whole flowgraph from the models of elements or subnetworks. This disadvantage is an essential limitation to the upward hierarchical analysis (see Section VI). The upward analysis makes possible the sequential analysis of subnetworks when the size of the whole network overflows the computer memory.

Two other methods of flowgraph formation, useful for decomposition analysis, are presented. Among different flowgraphs, representing the analysed network we should choose easily decomposable ones. It is better to avoid k-connected graphs [36] (with $k \geq 4$) because the topological analysis of such graphs, in spite of application of decomposition techniques, is very time-consuming.

The efficiency of computation is independent of whether the Mason or Coates graph representation is chosen. The Coates graph representation is more convenient for computer calculations because of the constant number of edges in a connection, for a given graph. This is the reason for the application of Coates representation in the methods of analysis presented in this paper.

Linear System of Equations and Its Coates Graph

In general, the Coates flowgraph can be formed not only for electronic circuit but for any linear system of equations having the form (2).

Definition 1 [19]

A Coates graph associated with a square matrix $\underset{\sim}{A} = [a_{ij}]$ being a matrix of linear system of equations

$$\underset{\sim}{A}\underset{\sim}{X} = \underset{\sim}{F} ,$$

where

$$\underset{\sim}{X} = [x_1, \dots x_n]^T, \underset{\sim}{F} = [f_1, \dots f_n]^T$$

is a flowgraph with n nodes. Nodes correspond to variables $x_1, \dots x_n$. A Coates graph edge goes from node x_j to node x_i having a weight equal to a_{ij}.

Coates graph associated with matrix $\underset{\sim}{A}$ will be denoted by $G_c(\underset{\sim}{A})$.

<u>Example 1</u>

Coates graph associated with the system of equations

$$\begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 0 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 8 \\ 0 \\ 9 \end{bmatrix} \qquad (7)$$

is shown in Fig. 2.

We have the following connections in this graph:

$$\{6,2,4\},\ \{7,1,3\},\ \{6,5,1\}$$

The following theorem, equivalent to the <u>Mason's rule</u> [78], can be concluded from the Binet-Cauchy theorem [36].

<u>Theorem 1 [19]</u>

The determinant of a matrix $\underset{\sim}{A}$ is given by

$$\det \underset{\sim}{A} = (-1)^n \sum_{p \in P} (-1)^{l_p} \det p\ , \qquad (8)$$

where

P is a set of connections of $G_c(\underset{\sim}{A})$,

l_p is the number of loops in connection p,

$\det p = \prod_{e \in p} y_e$, and y_e is a weight of an element e.

If weights of all elements are different there are no reducible terms in the dependence (8). det $\underset{\sim}{A}$ is sometimes called a <u>determinant of a graph</u> $G_c(\underset{\sim}{A})$, and det p a <u>determinant (weight) of a connection p</u>.

In an aim to solve a system of linear equations let us consider the following extended square matrix

$$\underset{\sim}{A}_u = \begin{bmatrix} m & \underset{\sim}{L} \\ -\underset{\sim}{F} & \underset{\sim}{A} \end{bmatrix}, \qquad (9)$$

where: $\underset{\sim}{A}$ and $\underset{\sim}{F}$ are as defined in (2) and m, $\underset{\sim}{L} = [\ell_1, \ell_2, \ldots \ell_n]$ consist of symbolic elements.

Matrix Coates graph $G_c(\underset{\sim}{A}_u)$ is shown in Fig. 3.

Let $\underset{\sim}{A}_i$ denote the matrix obtained by replacing the i-th column in matrix $\underset{\sim}{A}$ by the vector $\underset{\sim}{F}$.

Remark

Determinant of matrix A_u calculated from the formula (8) can be presented in the form

$$\det \underset{\sim}{A}_u = m \det \underset{\sim}{A} + \sum_{i=1}^{n} \ell_i \det \underset{\sim}{A}_i \quad . \tag{10}$$

Conclusion:

Taking into account equation (10) and Cramer's formulas, the solution for system of equations (2) with nonsingular matrix $\underset{\sim}{A}$ can be presented in the form

$$x_i = \frac{\det \underset{\sim}{A}_i}{\det \underset{\sim}{A}} = \frac{\sum_{p \varepsilon P_i} (-1)^{l_p} \det p}{\sum_{p \varepsilon P_m} (-1)^{l_p} \det p} \quad i=1,2,\ldots n \ , \tag{11}$$

where weights of additional edges m, $\ell_1, \ldots \ell_n$ are equal 1;

P_m - set of $G_c(A_u)$ connections enclosing edge m;

P_i - set of $G_c(A_u)$ connections enclosing edge ℓ_i.

Example 1 (cont.)

For the system of equation (7) matrix $\underset{\sim}{A}_u$ is of the form

$$\underset{\sim}{A}_u = \left[\begin{array}{c|ccc} m & \ell_1 & \ell_2 & \ell_3 \\ \hline -8 & 0 & 1 & 2 \\ 0 & 3 & 4 & 5 \\ 9 & 6 & 0 & 7 \end{array}\right].$$

The associated graph $G_c(\underset{\sim}{A}_u)$ is shown in Fig. 4. According to the formula (8), we have

$$\det \underset{\sim}{A}_u = -m\cdot 1\cdot 3\cdot 7 + m\cdot 1\cdot 5\cdot 6 - m\cdot 2\cdot 4\cdot 6 + \ell_1\cdot 8\cdot 4\cdot 7 + \ell_1\cdot 1\cdot 5\cdot 9$$
$$- \ell_1\cdot 2\cdot 4\cdot 9 - \ell_2\cdot 8\cdot 3\cdot 7 + \ell_2\cdot 8\cdot 5\cdot 6 + \ell_2\cdot 2\cdot 3\cdot 9 - \ell_3\cdot 1\cdot 3\cdot 9 - \ell_3\cdot 8\cdot 4\cdot 6 \qquad (12)$$

After ordering terms in (12) in the form of (10), any unknown variable can be easily calculated (as in (11)). For example,

$$x_3 = \frac{1\cdot 3\cdot 9 + 8\cdot 4\cdot 6}{1\cdot 3\cdot 7 - 1\cdot 5\cdot 6 + 2\cdot 4\cdot 6}.$$

If some of the unknown variables x_i are not to be calculated, the corresponding edges ℓ_i are removed from the extended graph.

Transitor Representation

As a convenient way of forming the signal flowgraph one can imagine the connection of models of such basic elements as two terminal and VCCS. These models can be obtained from the admittance equations. To reach this goal the notion of pseudo two-terminal transitor has been introduced.

Definition 2

A transitor is a pseudo two-terminal satisfying following equations (Fig. 5)

$$i_k = y\, v_m , \qquad (13)$$

$$i_m = 0 , \qquad (14)$$

where: v_m, v_k = potentials of nodes m and k

i_m, i_k = currents from nodes m and k

y = admittance (weight) of the transitor

In practice a transitor, as well as an unistor, can be described as a VCCS - Fig. 6a. In Fig. 6b, the unistor model for a transitor is shown. In Fig. 7, the transitor models for two-terminal and VCCS are presented.

Transitor models can be connected together resulting in the directed graph G with given weights of edges. The obtained graph has the structure similar to the structure of the analysed network.

Lemma 1

Coates signal flowgraph of analysed network can be obtained by removing from G one arbitrary node (treated as reference node) and all edges incident with this node [36].

The nodes of obtained graph correspond to nodal voltages with regard to the chosen reference node. The proof follows immediately from the form of admittance matrix of the network and submatrices representing transitors.

With an aid of models of two-terminal and VCCS the transitor model of any LLS network without operational amplifiers can be formed. Table I shows models of controlled sources obtained as an association of elements from Fig. 7.

Formal Transitors

When controlled sources of different kinds appear in the circuit, the application of models from Table I results in unnecessarily complicated flowgraph.

From the topological point of view, transitor is an edge of Coates flowgraph with a weight (transmittance) equal to the transitor admittance.

The graph is derived from the equations corresponding to the nodal analysis method. If we decide to derive our graph from the modified nodal analysis method [52], then variables may contain both nodal voltages and currents of controlled sources. Modified admittance matrix can be obtained as a sum of modified element admittance matrices. This property permits to form a flowgraph of a network by connecting graphs of its elements.

Definition 2

Graphs of network elements derived from modified nodal analysis method are called formal transitor models and their edges are called formal transitors.

In Table II formal transitor models for chosen network elements are shown. In equations describing the models in Table II, symbol i_x (x=a,b,c,d) indicates current from the node x.

It can be noticed that the structure of controlled sources models in Table II is less complicated than those of models in Table I.

Models of basic active elements of the linear electrical network can be obtained from the two-port model described with an aid of chain matrix $\underset{\sim}{A}$. When some elements of $\underset{\sim}{A}$ are equal to zero we remove from the model in position 8 (Table II) transitors with weights equal to 0. Let us consider the following examples:

a) formal transitor model of CCCS can be obtained when

$a_{11} = a_{12} = a_{21} = 0, \; a_{22} = -\frac{1}{\beta}$

b) converter model when $a_{12} = a_{21} = 0$

c) model of ideal operational amplifier can be obtained by setting

$a_{11} = a_{12} = a_{21} = a_{22} = 0.$

Generally speaking, models of two-ports that are not presented in Table II can be obtained by proper selection and adjusting of matrix parameters of general two-port formal transitor model.

Remark:

Models of ideal transformer and gyrator are equivalent respectively to the model of converter with $a_{11}\, a_{22} = 1$ and inverter with $a_{11}\, a_{22} = 1$.

Formal transitor model of VCCS obtained from a two-port model with chain matrix description (Fig. 8a) is slightly different from that obtained using admittance matrix description in Fig. 7b. Similarly, the CCVS obtained from a two-port with impedance description (Fig. 8b) differs from the model obtained from chain matrix description. In all other cases active element models obtained with an aid of different than chain matrix description have structure identical to that in Table II, and differ only in the weights of edges.

Generally any multi-terminal network having the following matrix form description:

$$\begin{bmatrix} \underset{\sim}{Y} & \underset{\sim}{A} \\ \underset{\sim}{B} & \underset{\sim}{Z} \end{bmatrix} \begin{bmatrix} \underset{\sim}{V} \\ \underset{\sim}{I}_o \end{bmatrix} = \begin{bmatrix} \underset{\sim}{I} \\ \underset{\sim}{0} \end{bmatrix}, \qquad (15)$$

where:

$\underset{\sim}{V}$ - vector of terminal voltages

$\underset{\sim}{I}$ - vector of terminal currents

$\underset{\sim}{I}_o$ - vector of some terminal currents

can have the formal transitor model with the matrix Coates graph shown in Fig. 9. Nodes of the Coates graph correspond to node voltages and some currents in terminals. Nodes I_o in transitor model are internal nodes. Model of multi-terminal network can be connected to other models with an aid of nodes V.

Example 2

Matrices $\underset{\sim}{Y}, \underset{\sim}{A}, \underset{\sim}{B}, \underset{\sim}{Z}$ for ideal operational amplifier are equal to

$$\underset{\sim}{Y} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad \underset{\sim}{A} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ -1 \end{bmatrix}, \quad \underset{\sim}{B} = [1\ -1\ 0\ 0], \quad \underset{\sim}{Z} = [0]$$

and vectors $\underset{\sim}{V}, \underset{\sim}{I}$ and $\underset{\sim}{I}_o$ are

$$\underset{\sim}{V} = [\underset{\sim}{v}_a, \underset{\sim}{v}_b, \underset{\sim}{v}_c, \underset{\sim}{v}_d]^T, \quad \underset{\sim}{I} = [i_a, i_b, i_c, i_d]^T$$

$$\underset{\sim}{I}_o = [i_o]$$

and according to the notation $i_o = i_c$.

Example 3

In Fig. 10, we have two multi-terminal networks each with the following system of equations:

$$\begin{bmatrix} \underset{\sim}{Y}_{11}^1 & \underset{\sim}{Y}_{12}^1 & \underset{\sim}{A}_1^1 \\ \underset{\sim}{Y}_{21}^1 & \underset{\sim}{Y}_{22}^1 & \underset{\sim}{A}_2^1 \\ \underset{\sim}{B}_1^1 & \underset{\sim}{B}_2^1 & \underset{\sim}{Z}^1 \end{bmatrix} \begin{bmatrix} \underset{\sim}{V}_1 \\ \underset{\sim}{V}_2 \\ \underset{\sim}{I}_o^1 \end{bmatrix} = \begin{bmatrix} \underset{\sim}{I}_1 \\ \underset{\sim}{I}_2^1 \\ \underset{\sim}{0} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \underset{\sim}{Y}_{11}^2 & \underset{\sim}{Y}_{12}^2 & \underset{\sim}{A}_1^2 \\ \underset{\sim}{Y}_{21}^2 & \underset{\sim}{Y}_{22}^2 & \underset{\sim}{A}_2^2 \\ \underset{\sim}{B}_1^2 & \underset{\sim}{B}_2^2 & \underset{\sim}{Z}^2 \end{bmatrix} \begin{bmatrix} \underset{\sim}{V}_3 \\ \underset{\sim}{V}_2 \\ \underset{\sim}{I}_o^2 \end{bmatrix} = \begin{bmatrix} \underset{\sim}{I}_3 \\ \underset{\sim}{I}_2^2 \\ \underset{\sim}{0} \end{bmatrix}. \quad (16)$$

Matrix Coates graph of the multi-terminal connection is presented in Fig. 11. Two subgraphs are connected in nodes V_2.

Remark

Grounding of any terminal causes the elimination of corresponding node from the graph and all edges incident with it.

When a flowgraph is obtained as a connection of element transitor models its nodes correspond to node voltages and some element currents. All other currents can be easily calculated and if necessary corresponding nodes can be taken into account in a graph structure.

For example if we are interested in a current in an admittance y, a current node should be connected to the existing voltage nodes as shown in Fig. 12. If we want to consider a current in a short circuit (as when calculating current or voltage-current transfer function), then a current node should be connected to the existing voltage nodes as shown in Fig. 13.

The models of elements shown in Table II are called autonomous models which means that when they are directly connected, they result in a flowgraph of the entire circuit. Adding a new element does not change the structure of an existing flowgraph. It is very convenient for forming the graph, but it is necessary to point out that the generated structure has still excess elements. When the internal currents are eliminated we obtain so-called nonautonomous models. When a non-autonomous model is connected to the flowgraph, the existing structure of a graph is to be changed. Changes are made in a set of edges incident with nodes of a new element and consist of eliminating and/or changing incidence nodes of some edges.

In Annex 1 the nonautonomous transitor models of electrical elements and directives for changes when these elements are incorporated to the flowgraph are presented.

It is to be noticed that changes caused by connection of nonautonomous models are easily implemented algorithmically. The use of nonautonomous models in the topological analysis can considerably reduce the computation time, especially when the analysis without decomposition is performed.

Transitor Models Derived from Tableau Equations

The tableau method of numerical analysis of linear networks [26] leads to system of equations with sparse matrix of coefficients. If sparse matrix techniques are applied to solve the above system of equations, good accuracy and speed of computations are achieved.

The characteristic feature of tableau equations is the surplus of independent variables which are currents and voltages of elements and nodal voltages. Tableau equations of a network with b edges and n nodes have matrix of coefficients with dimensions $(2b+n-1) \times (2b+n-1)$.

Formal transitor models derived from tableau description have no fundamental differences in comparison to those obtained from modified nodal approach. The essential difference lies in the model of two-terminal elements. In Fig. 14, the formal transitor model of two-terminal element derived from the tableau equations is shown.

In [65], the results of testing both two-terminal models are presented. Graphs obtained with the use of models in Fig. 14 have considerably less number of terms in its determinant than the one obtained with an aid of the model from Fig. 7. Nevertheless, the computation time was smaller when the models of Fig. 7 were used in comparison with the other representation because of the less number of flowgraph nodes.

Coates Graph of a Linear Transmitting Network

Having connected the transitor model of network elements the flow-graph should be completed with sources and auxiliary edges m and ℓ_1, ℓ_2, ... ℓ_n. In Fig. 15 the transitor models for independent current and voltage sources are presented. Node x_o is common for all independent sources (see Fig. 3) and is called a source node. The response in any graph node, current or voltage (except x_o) can be calculated for all sources simultaneously or for part of them only.

Remark: when superposition rule is to be used the edges J and -J or -E corresponding to the removed sources are removed from the graph.

Definition 4

A transitor graph of an electrical network is a Coates graph obtained by connecting transitor models of elements, completed with auxiliary edges m and ℓ_1, ℓ_2, ... ℓ_n and removing the reference node.

In the transitor graph the only considered edges ℓ_i can be those incident with nodes corresponding to currents and voltages to be calculated.

Example 4

Let us calculate the voltage transfer function

$$K_u = \left. \frac{u_4}{u_1} \right|_{i_4=0}$$

for the circuit shown in Fig. 16.

In Fig. 17 the transitor graph of this circuit containing a source u_1 is shown. The determinant Δ of transitor graph calculated from (12) is equal

$$\Delta = (-1)^7 \; [(-1)^4 \; m(Y_a+Y_b)(-1)(-Y_d) \; + \; (-1)^2 \; \ell_4(-v_1)(-Y_a)(Y_c+Y_d)$$
$$+ \; (-1)^2 \; \ell_4(-v_1)(-Y_c)(-1)(Y_a+Y_b)].$$

and so nodal voltage u_4 is equal to

$$u_4 = u_1 \; \frac{Y_a(Y_c+Y_d) - Y_c(Y_a+Y_b)}{Y_d(Y_a+Y_b)} \; ,$$

and the voltage transfer function

$$K_u = \frac{u_4}{u_1} = \frac{Y_aY_d-Y_cY_b}{Y_d(Y_a+Y_b)} \; .$$

Remark: the use of nonautonomous models (see Annex 1) gives for the above circuit a transitor graph with less number of edges and nodes (Fig. 18).

IV. TOPOLOGICAL FORMULAS

The notion of topological formulas is connected with the ability of calculation transfer functions of electrical networks with an aid of topological methods. The form of topological formulas is different for direct analysis and analysis with decomposition. It depends also on the type of partition and on the kind of topological representation. In practice the two-terminal immittances and two-port transfer functions are the most frequently calculated. In general, the ability of calculation of any matrix cofactor gives us the opportunity to have the description of any multi-terminal element.

As the basis for topological dependencies we consider evaluation of network immittances and transfer functions as functions of the determinant and cofactors of nodal admittance matrix.

Driving point admittance of a two-terminal (Fig. 19), with terminals p and q and indefinite admittance matrix Y, is equal to:

$$y = \frac{i(s)}{u(s)} = \frac{Y_{uv}}{Y_{pp,qq}} , \tag{17}$$

where

Y_{uv}, $Y_{pp,qq}$ - are cofactors of matrix $\underset{\sim}{Y}$ obtained by deleting columns and rows according to the indices (first index in a pair means row and second column)

u,v - chosen voltage and current reference nodes.

For a description of nondegenerated two-port it is sufficient to evaluate one of its matrices or generally (this covers also degenerated cases) characteristic polynomials [122].

The following formulas determine transfer functions of a two-port network presented in Fig. 20.

$$K_u = \left.\frac{u_2(s)}{u_1(s)}\right|_{i_2=0} = \frac{Y_{rp,sq}}{Y_{rr,ss}} , \tag{18}$$

$$K_{ui} = \left.\frac{i_2(s)}{u_1(s)}\right|_{u_2=0} = \frac{Y_{rp,sq}}{Y_{rr,ss,pp}-Y_{rr,ss,pq}-Y_{rr,ss,qp}+Y_{rr,ss,qq}} , \tag{19}$$

$$K_{iu} = \left.\frac{u_2(s)}{i_1(s)}\right|_{i_2=0} = \frac{Y_{rp,sq}}{Y_{uv}} , \tag{20}$$

$$K_i = \left.\frac{i_2(s)}{i_1(s)}\right|_{u_2=0} = \frac{Y_{rp,sq}}{Y_{pp,qq}} . \tag{21}$$

Characteristic polynomials of this two-port can be expressed as follows:

$$n_{oo} = Y_{uv} , \tag{22}$$

$$n_{os} = Y_{pp,qq} , \tag{23}$$

$$n_{so} = Y_{rr,ss} , \tag{24}$$

$$n_{ss} = Y_{rr,ss,pp} - Y_{rr,ss,pq} - Y_{rr,ss,qp} + Y_{rr,ss,qq} \; . \qquad (25)$$

In the case of three-terminal (q=s) the last formula is reduced to

$$n_{ss} = Y_{rr,ss,pp} \; . \qquad (26)$$

Additional polynomial (the transfer function denominator) is equal to

$$m(s) = Y_{rp,sq} \; . \qquad (27)$$

Second additional polynomial m_2 (denominator of transfer functions when the transmission in other direction is considered) which is related to others by equation $m_2 m = n_{oo} n_{ss} - n_{so} n_{os}$ can be calculated from the formula

$$m_2 = Y_{pr,qs} \; . \qquad (28)$$

Direct Topological Analysis

Equations (18) to (28) do not depend on the topological representation. They can be taken as a basis when topological formulas for different representations are established. It is also possible to derive topological formulas directly from transfer functions or characteristic polynomials definition.

Let us denote W a set of pairs of nodes in the Coates graph G_c:

$$W = \{(v_1, r_1), \dots (v_k, r_k)\}, \; v_l \neq v_m, \; v_l \neq r_m, \; r_l \neq r_m \text{ for } l \neq m.$$

Definition 5

We call a k-connection (multiconnection) of a graph G_c a subgraph p_W, composed of k separated paths and disjoint with them loops incident with all graph nodes. The starting node of i-th path is v_i and the terminal node is r_i (pairs of nodes from the set W).

In Fig. 21 a flowgraph and its 1-connection p_W is presented. In this case W = {(5,1)}. 0-connection or simply connection is denoted by p, because W=∅. When $v_i = r_i$ the multiconnection has the isolated node

v_i.

Multiconnection is natural generalisation of terms "connection" and "1-connection" defined by Coates [29] and is useful for the topological analysis of decomposed network. This notion corresponds to that of k-tree (multitree) occuring in the analysis (with decomposition) when the representation with pair of conjugated graphs or directed graph is used.

A tree can be obtained from the k-tree, by adding k-1 edges, analogically k-connection can be transformed into connection by adding k edges. A set of all k-connections p_W will be denoted by P_W.

Definition 6

Weight function $|P_W|$ of multiconnection set P_W of Coates graph with n nodes is defined as follows:

$$|P_W| = \sum_{p \varepsilon P_W} \text{sign } p \prod_{e \varepsilon p} y_e \ , \tag{29}$$

where $\text{sign } p = (-1)^{n+k+l_p} \text{ ord } (v_1,\dots,v_k) \text{ ord } (r_1,\dots,r_k)$,

$$\text{ord } (x_1, x_2, \dots x_k) = \begin{cases} 1 \text{ when the number of permutations} \\ \quad \text{ordering the set is even,} \\ 1 \text{ in the opposite case,} \end{cases}$$

l_p - number of loops in multiconnection p ,

y_e - weight of element e.

Consider a transitor graph of a two-port network in Fig. 20 with removed reference nodes.

It can be demonstrated that cofactors of admittance matrix are equal:

$$Y_{uv} = |P| \ , \tag{30}$$

$$Y_{rp,ss} = |P_{\{(r,p)\}}|, \quad r \neq s, \ p \neq s \ , \tag{31}$$

$$Y_{pq,rr,ss} = |P_{\{(p,q),(r,r)\}}|, \; p \neq s, \; q \neq s, \; r \neq s, \; p \neq r, \; q \neq r \; . \qquad (32)$$

With an aid of above formulas we have the following result.

Result 1

Characteristic polynomials of two-port represented with an aid of a flowgraph can be evaluated from the following topological formulas

$$n_{oo} = |P| \; , \qquad (33)$$

$$\begin{aligned} n_{os} &= Y_{pp,qq} = Y_{pp,ss} + Y_{qq,ss} - Y_{pq,ss} - Y_{qp,ss} \\ &= |P_{\{(p,p)\}}| + |P_{\{(q,q)\}}| - |P_{\{(p,q)\}}| - |P_{\{(q,p)\}}| \; , \end{aligned} \qquad (34)$$

$$n_{so} = |P_{\{(r,r)\}}| \; , \qquad (35)$$

$$\begin{aligned} n_{ss} &= |P_{\{(r,r),(p,p)\}}| - |P_{\{(r,r),(p,q)\}}| + \\ &\quad - |P_{\{(r,r),(q,p)\}}| + |P_{\{(r,r),(q,q)\}}| \; , \end{aligned} \qquad (36)$$

$$m = Y_{rp,sq} = Y_{rp,ss} - Y_{rq,ss} = |P_{\{(r,p)\}}| - |P_{\{(r,q)\}}| \; . \qquad (37)$$

In formulas (30) to (37) P_W denotes set of respective multiconnections of a graph G_c with symbols as in Fig. 20.

When the direct topological analysis is performed it is more convenient to operate with topological formulas using the connection concept only. In this case the computer realization needs only the generator of connections. The above demand can easily be satisfied in a similar way in the case of solving linear equations (Sec. III). The auxiliary edges (a,b,c,d,e,f,g) with weights -1 are connected to the flowgraph as schematically shown in Fig. 22 [65].

If P is a set of all flowgraph connections and P^x means its subset containing only connections with an edge x, then characteristic polynomials of a two-port can be evaluated as follows:

$$n_{oo} = |P| \; ,$$

$$n_{os} = |P^b| + |P^c| - |P^f| - |P^e| \; ,$$

$$n_{so} = |P^{a}| \ ,$$ (38)

$$n_{ss} = |P^{a,b}| + |P^{a,c}| - |P^{a,f}| - |P^{a,e}| \ ,$$

and

$$m = |P^{d}| - |P^{g}| \ .$$ (39)

Computer programs SNAP and NASAP [26] are based on direct topological analysis of flowgraph representing the electrical network. These programs can be used for networks having no more than 30 nodes (in practice 10 nodes, 30 edges). This is due to the great number of terms in direct topological formulas. Computer time needed for realization of direct topological formulas is proportional to the number of connections in a flowgraph.

Let $\underset{\sim}{D}(G) = [d_{ij}]$ be a matrix denoting the connection of a Coates flowgraph; d_{ij} is equal to the number of edges directed from the node i to the node j. $\underset{\sim}{D}$ is a square matrix with the dimension equal to the number of nodes. Number of connections in a graph is equal [20]

$$\text{card } P = \text{per } [D(G)] \ ,$$ (40)

where per $\underset{\sim}{A}$ is a permanent of matrix $\underset{\sim}{A}$ [19].

The very rough estimation for the number of connections for the graph with n nodes and k edges is [111]

$$\text{card } P \leq \left(\frac{k+1}{n} - 1\right)^{n} \ .$$ (41)

Although (41) is only an upper estimation, it expresses correctly the characteristic of changes in the number of terms. The exponential increase of number of terms is observed in practice for the direct topological analysis, which causes that direct analysis of large networks is inexecutable.

V. THE GRAPH DECOMPOSITION

The graph of electrical network can be analysed directly with an aid of formulas (38), (39) and the transfer function of analysed network can be obtained in all symbolic form. From the previously presented discussion, it is evident that the number of terms in symbolic function is too large. This is the reason why the analysis of medium and large networks is a formidable task and very often inexecutable. This introduces the necessity of a network decomposition and consequently the graph decomposition.

Types of Partition

Procedure of graph partition and determination of parts called blocks will be called decomposition.

A flowgraph can be decomposed in one of three manners:

1. Node decomposition. A graph is divided into edge disjoint subgraphs (blocks) (Fig. 23). Nodes common for two or more blocks are called block nodes. A notion of block graph [123] is connected with this type of decomposition. Block graph is exploited in the topological analysis and algebra of the second category structural numbers [6]. A node decomposition can be used for all types of network representation. A particular case of node decomposition is bisection or decomposition into two subgraphs.

2. Edge decomposition. In a graph we isolate a node disjoint blocks. Blocks are connected together by the means of edges which form cutsets

of the graph (Fig. 24). These edges are called cutting edges. In the case of edge decomposition nodes incident with cutting edges are called block nodes.

3. Hybrid decomposition. This partition is a combination of two previous decompositions (Fig. 25). Nodes incident with cutting nodes or common for more than one block are block nodes.

In both edge and hybrid decompositions, a bisection can be distinguished as a special case.

We can focus our attention on bisection because of its special usefulness for the hierarchical decomposition. It is evident that any decomposition can be represented as a sequence of bisections, and for computer algorithms such assumption produces simple data structures and organization of computations.

Definition 7

Complete symmetrical directed graph with self loops spanned on block vertices of subgraph G_i is called a substitute graph for that block and is denoted G_i^s (Fig. 26).

Definition 8

Graph G^d obtained when replacing blocks G_i by their substitute graphs is called a decomposition substitute graph.

In the case of edge or hybrid decomposition, cutting edges belong to the decomposition substitute graph (Fig. 27).

A decomposition substitute graph should not be too complex because cost of its analysis depends on the number of edges and nodes exactly as estimated for the case of proper graph (40), (41). Hence, it appears necessary to limit the number of blocks and block nodes. This limitation causes uneffectiveness of simple decomposition method for the case of very large networks. For large networks either decomposition substitute graph G^d is too complex for analysis or blocks G_i are still too complex for direct topological analysis.

When simple decomposition is applied to subgraphs, we deal with hierarchical decomposition.

Algorithms of Graph Partition

Decomposition of network graph should be executed automatically. There are two reasons for that: firstly, the graph structure is not known when network data are provided and a priori decision about block partition regarding only network structure could be nonoptimal, and secondly elaborating of data would be cumbersome for the program user and would demand the knowledge of decomposition methods and calculations regarding whether the partition is profitable or not.

There is a lot of graph decomposition methods and they can be classified in four main groups:

1) isolating of k-connected subgraphs (cliques) [40,53,75],
2) exchanging of nodes among two subgraphs, up to the moment when local minimum of quality index is reached [57],
3) solving equivalent system of mathematical equations [45],
4) finding a graph contour [93,100,109].

Any of the above methods does not provide us with the efficient algorithm that gives optimal solution (global minimum). Only part of them has efficient heuristic algorithms.

Problem of graph decomposition is of the non-polynomially bounded class which means that the time τ of finding an optimal decomposition cannot be limited by a polynomial of nodes (w) or edges (k) number.

Taking the above into account we should not expect efficient algorithm giving optimal solutions. Investigations should be performed for finding algorithms providing correct and nearly optimal solution in a short time.

One of such efficient algorithms has been presented in [100], and a modification of it proposed in [109] gives better partitions than [100] with the simultaneously improved efficiency of computations. The time of graph decomposition by the aforementioned algorithm depends linearly on the number of nodes.

VI. HIERARCHICAL ANALYSIS

Let us consider the case of node decomposition.

In Fig. 28 an example of hierarchical decomposition is presented. The hierarchical decomposition structure can be illustrated by a tree of decomposition. Nodes of the tree correspond to subgraphs obtained on different levels of decomposition. If a subgraph G_k was obtained during decomposition of subgraph G_i, then there is an edge from G_i node to G_k node. Fig. 29 shows the tree of decomposition from Fig. 28.

In the decomposition tree we have one <u>initial node</u> which is only the starting point of edges. <u>Terminal nodes</u> are those which are only the ending points of edges. All nodes that are not terminal nodes are <u>middle nodes</u>.

For middle nodes we determine <u>decomposition level</u> which is equal to the number of nodes in the path from the initial node to that node. Range of hierarchical decomposition is equal to the maximal decomposition level.

Blocks associated with terminal nodes are called proper blocks. Every middle node has its descendants and every node except the initial one has its ascendant.

If we limit ourselves to the bisection as the only graph partition, every middle node has exactly two descendants. As remarked previously, every decomposition can be considered as a sequence of bisections in hierarchical structure. Hence, in further consideration and without a loss of generality, we shall examine this case only. This results in a simpler form of formulas and easier algorithm organization.

When the hierarchical decomposition analysis is to be performed, the following problems have to be solved:

a) direct topological analysis of proper blocks (terminal nodes),

b) analysis of middle blocks which consists of how to combine results from the lower level of decomposition to get description of a subgraph on the higher level.

Analysis of Proper Block

Let us consider a connection of Coates graph evaluated when the topological formulas are realized. When we deal with decomposed graph we can see that the part of the connection contained in a particular proper block, forms a multiconnection in this block.

The incidence of the block nodes notified in the set W determines the kind of multiconnection. It means that topological analysis of proper blocks will consist of enumeration of multiconnections, with paths linking different block nodes. Analysis on middle nodes level will consist in combining together various kinds of multiconnections.

It is evident that combining multiconnections one by one will not reduce considerably the computation time. Multiconnections should be generated in groups and whole groups should be combined together. The larger the groups of multiconnections are the simpler the proper block analysis is, and the more efficient intermediate level analysis is. One rule should be obeyed, namely, the resulting multiconnections should be generated without duplications.

Therefore it is to be decided which kind of multiconnection type characterization should be chosen.

The most detailed characterization is that presented in definition 5, which is the generalization of Coates definition of 0- and 1-connections. For a block the different multiconnections may be grouped in sets P_W, of multiconnections characterized by the same set of nodes W.

However, it should be noted that block with nb block nodes has

$$M(nb) = \sum_{i=0}^{nb} \binom{nb}{i}^2 i! \qquad (42)$$

different types of multiconnection sets. This dependence could seriously limit the decomposition method. This led us to investigate other characterizations of multiconnection set. After some trials [65, 66] the following type of characterization was chosen.

<u>Definition 9</u>

P(B,E) is a set of multiconnections which have the following properties:

a) the only incidence of block nodes is defined by sets B, E - nodes of B are origins and E are ends of multiconnection edges,

b) all other nodes (internal nodes of a block) have full incidence,

where $B = \{b_1, b_2, \ldots, b_m\}$, $E = \{e_1, e_2, \ldots, e_m\}$ and $B \cup E \subset NB$ - set of block nodes.

Remark 1 Block nodes which are not included in $B \cup E$ are isolated nodes.

Remark 2 In the sense of this definition in the set P(B,E) we can find k-connections with $k = \text{card}(B - B \cap E) = \text{card}(E - B \cap E)$.

Example 5

Let us consider the block shown in Fig. 30. If NB = {1,2,3,4}, B = {1,2,}, E = {3,4}, the set P(B,E) is equal to {{1,5,3}, {2,4,6}}. According to the definition 5, each of these multiconnections has different characterizations by sets W.

$$\{1,5,3\} \in P\{\{1,4\}, \{2,3\}\},$$

$$\{2,4,6\} \in P\{\{1,3\}, \{2,4\}\}.$$

The weight function of multiconnection set P(E,B) is defined as in (29). It is to be noted that for a block with nb block nodes, the number of different types of multiconnection sets P(E,B) is

$$MR(nb) = \Sigma \binom{nb}{i}^2 \qquad (43)$$

which means an important reduction in comparison with (42). It will be shown that at the same time formulas can be elaborated permitting generation of connections without duplication with the use of their new characterization.

Analysis of Middle Block

Analysis on intermediate level consists of evaluation of multiconnection of a block that results from the association of two (or in general more) blocks. Let us denote the sets of block nodes for both blocks and the resulting block by NB_1, NB_2 and NB respectively. When connecting two blocks, some of their block nodes become internal nodes, which means that no other blocks are connected to these nodes on upper levels. These nodes will be called reducible nodes.

Let us denote

$$COM = NB_1 \cap NB_2 \text{ - the set of common nodes ,} \quad (44)$$
$$RED = COM - NB \text{ - the set of reducible nodes ,}$$

$P_1(B_1,E_1)$, $P_2(B_2,E_2)$ and P(B,E) - the design sets of multiconnections for both blocks and resulting block, respectively.

Theorem 2

Any set of multiconnections P(B,E) can be obtained according to the following rule

$$P(B,E) = \bigcup P_1(B_1,E_1) \times P_2(B_2,E_2) \ , \quad (45)$$

where summation is performed over all sets of multiconnections P_1 and P_2 satisfying conditions:

$$B_1 \cap B_2 = \emptyset, \ E_1 \cap E_2 = \emptyset \ , \quad (46)$$
$$RED \subset (B_1 \cup B_2) \cap (E_1 \cup E_2) \ .$$

Sets B and E are in this case equal to:

$$B = B_1 \cup B_2 - RED, \ E = E_1 \cup E_2 - RED.$$

If all element weights are different, there are no duplicate terms in the formula (45). For every multiconnection $p \in P$, the sign of p can be calculated as follows

$$\text{sign } p = \text{sign } p_1 \cdot \text{sign } p_2 \cdot (-1)^k \cdot \Delta \qquad (47)$$

where

$$p = p_1 \cup p_2, \quad p_1 \in P_1, \; p_2 \in P_2$$

$$k = \min(\text{card}(E_1 \cap B_2 \cap COM), \text{card}(E_2 \cap B_1 \cap COM)) + \text{card}(COM)$$

$$\Delta = \text{ord}(b_{11}, b_{22}, \ldots, b_{1m_1}) \text{ ord}(e_{11}, e_{12}, \ldots, e_{1m_1})$$
$$\text{ord}(b_{21}, b_{22}, \ldots, b_{2m_2}) \text{ ord}(e_{21}, e_{22}, \ldots, e_{2m_2})$$

$$B_1 = \{b_{11}, b_{12}, \ldots b_{1m_1}\}, \quad E_1 = \{e_{11}, e_{12}, \ldots e_{1m_1}\},$$

$$B_2 = \{b_{21}, b_{22}, \ldots b_{2m_2}\}, \quad E_2 = \{e_{21}, e_{22}, \ldots e_{2m_2}\}.$$

Remark An important feature of (45) is the possibility of obtaining a set of multiconnections P(B,E) by combining whole groups of multiconnections from the lower level. At the same time, from (47) we can notice that the new sign is attributed simultaneously to the whole group of terms $P_1 \times P_2$, as k and Δ depends only on sets B_1, E_1, B_2, E_2. These features are of great importance in the computer realization because we do not have to deal with each multiconnection on a separate basis.

Example 6

Let us consider an association of two blocks presented in Figs. 31 and 32.

$NB_1 = \{1,2,4\}$, $NB_2 = \{2,3,4\}$, $NB = \{1,2,3\}$, $COM = \{2,4\}$, $RED = \{4\}$.

Let us calculate multiconnections of the type P({1,2}, {2,3}) of the resulting block. From the formula (45), with the condition (46), we obtain

$$P(\{1,2\}, \{2,3\}) = P_1(\{1,4\}, \{2,4\}) \times P_2(\{2\}, \{3\}) \cup$$
$$P_1(\{1\}, \{2\}) \times P_2(\{2,4\}, \{3,4\}) \cup$$
$$P_1(\{1,2\}, \{2,4\}) \times P_2(\{4\}, \{3\}) \cup$$
$$P_1(\{1\}, \{4\}) \times P_2(\{2,4\}, \{2,3\}).$$

A formula similar to that of Theorem 2 can be derived for the case of edge decomposition. Analysis of proper blocks is realized in the same way as described previously. An edge bisection will be considered. We denote:

E_{cut} - cutset of a graph G,

$G_1(E_1,V_1)$, $G_2(E_2,V_2)$ - two disconnected graphs obtained from G after removing edges E_{cut},

NB, NB_1, NB_2 - sets of block vertices for G, G_1 and G_2 respectively,

$RED = NB_1 \cup NB_2 - NB$,

P_{cut} - set of multiconnections formed only by edges E_{cut}.

<u>Theorem 3</u> [65]

Any set of multiconnections P(B,E) can be obtained according to the following rule*

$$P(B,E) = \bigcup P_1(B_1, E_1) \times P_2(B_2, E_2) \times P_{cut}(B_c, E_c) , \qquad (48)$$

where summation is performed over all sets of multiconnections P_{cut}, with sets B_1, E_1, B_2, E_2 satisfying the following conditions:

$$B_c \cap B_1 = B_c \cap B_2 = E_c \cap E_1 = E_c \cap E_2 = \emptyset$$

$$RED \subset (B_c \cup B_1 \cup B_2) \cap (E_c \cup E_1 \cup E_2)$$

Sets B and E are then equal to

---

* This form of the formula (48), being a modification of the formula presented in [65], have been proposed by M. Bonn.

$B = B_1 \cup B_2 \cup B_c - RED; E = E_1 \cup E_2 \cup E_c - RED.$

If all element weights are different, there are no duplicate terms in the formula (48). For every multiconnection $p \in P$, the sign of p can be calculated as follows:

$$\text{sign } p = \text{sign } p_1 \cdot \text{sign } p_2 \cdot \text{sign } p_{cut} \qquad (49)$$

Downward and Upward Hierarchical Analysis

When the method of analysis of terminal blocks and middle blocks is elaborated, it remains to organize the exploration of hierarchical structure to obtain description of the initial network.

Two approaches are possible and are called downward and upward method of analysis. The name is connected with the direction of exploration of the decomposition tree as represented in Fig. 29.

The upward method presents many advantages over the downward method for algorithm organization, saving of computer time and memory, so the later will be only briefly outlined.

In the downward method, the analysis starts from the 1-level (initial block) and proceeds down to the next levels according to connections in the hierarchical tree. The substitute graphs of blocks corresponding to middle nodes are analysed. On each intermediate level the type of necessary functions from the next level is determined. Arriving to the terminal node the analysis of the subgraph is executed to get the necessary function of this block. Then the way up is performed. For each passing by the middle node, the multiplication of two functions from the lower level is executed. When the way back is accomplished, we obtain a part of the function of the initial network. In fact, many down and up processing have to be performed. The formula

(45) expresses a set of multiconnections of the middle block as a sum of products of multiconnection sets from the lower level. Each term of this sum requires the described above up and down procedure.

The downward method presented in [65] permitted the hierarchical analysis of large networks but at the same time had some disadvantages. These disadvantages consisted mainly of:

(a) multiple passes through the hierarchical structure which cause multiple calculations of the same function,

(b) complicated organization schema and some practical problems with efficient storage of all-symbolic results.

For these reasons a new form of hierarchical tree exploration was elaborated. In the new method, only one passing along the hierarchical structure is necessary. The name of upward method is due to the direction of exploration.

Let us describe the process more in detail. Firstly, to facilitate the further organization of the algorithm, a specific numeration of blocks is introduced. If N is the number of blocks (i.e. terminal and middle nodes), we shall number them from 1 to N. The only condition for that numeration is that ascendant should have lower number than its descendants. Such a numeration is easy to be performed, e.g., we can numerate nodes starting from the 1-level and sequentially to the lowest level (as in Fig. 29). With this numeration the initial block has always number 1.

The upward method of analysis starts from the block having the number N and is performed sequentially to number 1. When the terminal node is met, the analysis that is presented in Section VI is executed. When we arrive to the middle block, where its descendants have been previously analysed, the formula (45) is executed. Two approaches are possible:

(a) using the substitute graph, the combinations fulfilling conditions of the Theorem 2 are found,

(b) after examination of all possible combinations of descendant blocks functions only those fulfilling conditions of the Theorem 2 are retained.

Since simple test of combinations has been found (see Section VII) the second solution was chosen for the algorithm and the program. The procedure ends when the initial block is analysed. At that moment the functions of initial networks are calculated.

VII. ALGORITHM OF UPWARD HIERARCHICAL ANALYSIS

As can be noted from the general presentation of the method, there are two distinct parts to be executed: analysis of terminal blocks and analysis of middle blocks. These two problems are resolved separately and each part can be ameliorated without affecting the other.

Analysis of Proper Block

The algorithm of generation of multiconnections of Coates graph will be presented. This part of the method corresponds to the methods of direct topological analysis of electrical circuits. The generation of 0-connections of flow-graph can be converted to the problem of generation of disjoint cycles of a graph (see [26,50]).

Among different methods of generation we should notice techniques consisting of the generation of cycles of the graph and sets of disjoint cycles, and afterwards finding all O-connections of the graph. The technique presented in [26] has been applied in program SNAP [74]. The disadvantage of the method is the necessity of storing in the computer memory all cycles and all sets of disjoint cycles. Other methods of cycle generation can be found in [43,118,124,127]. Now the algorithm used in the program of hierarchical analysis will be presented.

Let us consider a Coates graph with n nodes. Let $\underset{\sim}{M}$ be an incidence matrix defined as follows

$\underset{\sim}{M} = [m_{ij}]_{n x n}$; m_{ij} = set of edges starting from the i^{th} node and ending at the j^{th} node.

Set of O-connections of a flow-graph can be calculated from the formula

$$P = \bigcup_{(i_1,\ldots,i_n) \in I} m_{1i_1} \times m_{2i_2} \times \ldots \times m_{ni_n}, \qquad (50)$$

where I - set of all permutations of numbers (1,2,...n). There is no duplication in the formula (50). A sign of O-connection $p \in m_{1i_1} \times m_{2i_2} \times \ldots m_{ni_n}$ is equal to number of permutations necessary to order set $i_1, \ldots, i_n$ multiplied by $(-1)^n$.

In the formulas for the hierarchical analysis, not only set of all O-connections is necessary but also sets of multiconnections characterized in Definition 9. This problem can be easily transformed to the generation of all O-connections of the modified graph.

Lemma 2

The set of multiconnections P(B,E) of a graph with an incidence matrix $\underset{\sim}{M}$ is equal to the set of O-connections of a graph described by a matrix $\underset{\sim}{M}(B,E)$. The matrix $\underset{\sim}{M}(B,E)$ is obtained from the matrix $\underset{\sim}{M}$ by deleting:

- all columns corresponding to nodes from B,

- all rows corresponding to nodes from E.

The complete description of the block with nb block nodes is covered by weight functions of all possible sets P(B,E) with $B \cup E \subset NB$. The different sets B,E can be generated in the manner described below.

Let us numerate nodes of NB from 1 to nb. For i=0,...,nb, all i-element subsets of the set {1,...,nb} are generated. For a given i, let $\binom{nb}{i}$ such subsets form a set K(i). To each pair m, k ε K(i) (note that m may be equal k) corresponds a set of potential multiconnections P(B,E) with B=m and E=k. Such sets of multiconnections are generated and stored. Each set may be identified by its type B,E. This type may be coded on a one computer word. The 2*nb bits would be occupied. Successive pairs of bits describe block nodes from 1 to nb. All elements b from B produce 1 on position 2*b-1 and elements e from E produce 1 on position 2*e. All other positions are equal to 0. The code C of a set of multiconnections P(B,E) can be thoroughly calculated from the formula

$$C = \sum_{b \varepsilon B} 2**(2b-2) + \sum_{e \varepsilon E} 2**(2e-1) . \qquad (51)$$

Example 7

For the set of block nodes NB = {1,2,3,4} 8 bits will be occupied to code different sets of multiconnections. If B = {1,2} and E = {2,3} the code for P(B,E) will be equal

$$C = 2^0 + 2^2 + 2^3 + 2^5 = 45.$$

This coding permits an easy identification of multiconnection set by one integer number and furtherwards a simple practical realization of formula (45).

Analysis of Middle Block

In the upward hierarchical method, the analysis of middle block is performed at the moment when both its descendants have been analysed. The sets of multiconnections of these blocks are stored in the computer memory each having an identification code.

Let us follow certain rules of block nodes numeration (execute renumeration if necessary):

- first group is formed of reducible nodes RED
- second group is formed of other common nodes COM-RED
- third group is formed of other block nodes.

Both first and second group should have the same numeration in blocks to be associated.

We examine all possible combinations of functions describing two blocks. Let us denote the following bit fields in a computer word containing the code of a multiconnection:

RED_1, RED_2 - corresponding to the nodes RED in both blocks (first group),

CR_1, CR_2 - corresponding to the second group of nodes,

$REST_1$, $REST_2$, - corresponding to third group of nodes.

The following tests are executed

$$\begin{aligned} &\text{AND } (RED_1, RED_2) = 0, \\ &\text{AND } (CR_1, CR_2) = 0, \qquad (52) \\ &\text{OR } (RED_1, RED_2) = \text{field having 1 on each bit.} \end{aligned}$$

For the chosen code of multiconnection (51) conditions (52) are equivalent to (46). So if any of these conditions is not fulfilled, the combination is rejected. In the contrary case, we retain the combination, which is characterized by sets of nodes according to the formulas (46).

The code for resulting multiconnections can easily be composed from parts of codes of component multiconnections. As the first group of nodes is no more block nodes, there is no information concerning this group. Nodes from the second group have code equal $OR(CR_1, CR_2) = CR_1 + CR_2$. As nodes from the third group are distinct in two blocks, their description remains $REST_1$, $REST_2$.

General Organization of Algorithm

General organization of the algorithm is presented in the Table III. Once the proper numeration of block nodes is established, the analysis can be carried out as presented. In fact with this numeration, analysis of any middle block is performed when both his descendants have been analysed. The last analysed block is the initial block.

The all-symbolic or semi-symbolic descriptions for large networks are only intermediate results which are elaborated further with the object of performing different types of network analysis.

Symbolic form of transfer function for large network contains a very large number of terms. To make possible the storage and to facilitate further elaboration decomposed form of results will be preserved. This form was proposed in [14] and it is particularly convenient for the upward hierarchical method.

Description of a terminal block is formed of characteristic functions as presented in section IV. Each term of function has the form

$$t = r \cdot s^k \prod_i y_i \tag{53}$$

where: r - numerical factor; s - Laplace variable; y_i - symbolic admittances or symbolic element parameters.

Any characteristic function is stored in the form of three vectors with successive elements equal: r, k, coded y_i. Each different function can be identified by its code (51).

From formula (45) we see that any function for a middle block is expressed as a sum of products of functions from the lower level. For upward hierarchical method, the analysis of any middle block is performed, when two blocks being its descendants have been previously analysed and resulting functions stored. The function of middle block can be stored in an unexpanded form containing only addresses of functions to be multiplied. A term of such function is of the form:

$$m = v \cdot F(i) \cdot F(k) \quad , \tag{54}$$

v - sign of term equal $\pm$ 1,

F(i), F(k) - functions describing descendants of analysed block.

The term m can be memorized by three numbers: v and addresses of F(i) and F(k) stored previously.

The analysis is terminated by analysing the initial block. To profit of these results the whole structure should be run through. From the functions of initial block we only choose the necessary ones. The given addresses send us to next blocks. At the end we find functions of terminal block. On these functions different kinds of operations can be performed, depending on what kind of analysis is requested.

Example 8

Let us take a practical network to illustrate the successive parts of the algorithm. In Fig. 33 the scheme of analysed band-pass filter is presented. Operational amplifiers are considered ideal. The Coates flowgraph corresponding to this network can be found in Fig. 34. This graph has been decomposed into 5 terminal blocks (Fig. 35). The hierarchical structure of successive associations is presented in Fig. 36. Symbolic results for the network are in Table IV. The results are presented in the unexpanded form as they are computed by the program. The voltage transfer function for the considered filter can be expressed as

$$K_u = \frac{F(24)}{F(25)} \tag{55}$$

Examining once again the whole structure, the symbolic transfer function can be found. They can be exploited in compact form or expanded if necessary. This network has 44 elements and consequently 44 symbolic parameters in symbolic results. All symbolic analysis of networks of this size can take quite a considerable computer time when direct topological methods are applied. In the case of hierarchical analysis, it is even possible to obtain these results by hand calculations.

It can be noticed that for this specifical structure graphs of 4 blocks: block 9,8,7,6 are isomorphic. If an isomorphism of graphs is detected, it is possible to execute block analysis only once because the symbolic description of both blocks is identical. This permits to make savings in computer time and the memory needed for storing the results.

VIII. COMPUTER PROGRAM OF HIERARCHICAL ANALYSIS

Technical Description of Programs

Two computer programs were realized on the basis of presented algorithms. Program FANES [65] realizes the downward analysis of hiearchical structure. The edge decomposition is used in this program. Some comparisons between SNAPEST, NAPPE, SNAP [74,121] and FANES are presented in [65].

First results of the program FLOWUP realizing the upward hierarchical method were published in [66]. Program FLOWUP is written in FORTRAN and is implemented on CDC Cyber 73 and CIIHB DPS/8 computers. Memory demands for the program are not important and additionally two parts of the program, namely, terminal and middle block analysis can be separated and overlayed. The BASIC version for the minicomputer HP9835 (or HP9845) with standard memory was realized.

For the terminal block analysis matrices of the range nxn are to be stored. Where n-number of block nodes (in general not greater than 10). The demand for middle block analysis is due to the number of block nodes. In the case of large networks analysis the most important memory demand is connected with the storage of results. Three vectors (53), each with length equal to the number of terms are necessary. As the compact form is used the all-symbolic form for quite large networks can be calculated. In the minicomputer version the successive transfer of results to the other memory supports may be performed during the execution of the program.

Results and Comparisons

Let us present now some comparative results of analysis with the FLOWUP program.

At first let us examine the ladder structure decomposed into different terminal blocks as shown in Fig. 37. Time of computer analysis and number of terms in results are presented in Figs. 38 and 39. The isomorphism of terminal blocks was not exploited. Both time and memory depends linearly on the number of nodes of analyzed ladder. Linear dependence is characteristical for all cascade connections of blocks. It can be noticed that both time and memory depends on the kind of partition performed. These computations have been performed on CDC Cyber 73.

Analysis of the filter presented in Fig. 40 was executed on DTS/8 GCOS. Time of analysis of this filter was 0.165 sec. In the case of cascade connection of Sections we have the linear growth of computer time as presented in Fig. 41. The isomorphism of sections has not been taken into account. When connections of block is more complicated than cascade the analysis is expected to be more time consuming.

IX. CONCLUSIONS

We discussed new method that increases computational power of topological analysis due to the reduction in the computer time needed for the analysis of large electronic networks. The approach will significantly affect the applications of topological methods to the analysis of large networks, which was impossible, even with the help of the fastest computers.

New possibilities of using the topological analysis towards solution of network design problems, with the help of the fully symbolic form of results, arise. They can be used for sensitivity analysis, tolerance analysis and approximate symbolic analysis [115,116].

The method preserves the advantages of direct methods of topological analysis such as high accuracy of computations and possibility of generating fully symbolical results.

Besides, any system of liner equations can be analyzed symbolically after representing it in the form of a Coates flowgraph.

The restriction of the presented method in its application to large networks lies in the number of block nodes in each block. This is usually overcome by using effective decomposition algorithm which minimizes the number of partition nodes.

REFERENCES

[1] C. Acar, "New expansion for signal-flow graph determinant", Electr. Letters, Dec. 1971.

[2] G.E. Alderson and P.M. Lin, "Computer generation of symbolic network functions - a new theory and implementation", IEEE Trans. Circuit Theory, Vol. CT-20, pp. 48-56, Jan. 1973.

[3] S. Bellert, "Computer four-pole synthesis based on the method of structural numbers", Arch. Elektr., t. XIII, z. 3, 1964.

[4] S. Bellert, "Topological considerations and synthesis of linear networks by means of the method of structural numbers", Arch. Elektr., t. XII, z. 3, 1963.

[5] S. Bellert, "Topological analysis and synthesis of linear systems", J. Franklin Institute, vol. 274, Dec. 1962.

[6] S. Bellert and H. Wozniacki, Analiza i synteza ukladow elektrycznych metoda liczb strukturalnych, Warszawa: WNT, 1968.

[7] A.V. Bondarenko, "Nullator-norator models for controlled voltage and current sources", Telecommun. Radio Eng., vol. 28/29, 1974.

[8] F.H. Branin, "A sparse matrix modification of Kron's method piecewise analysis", IEEE Int. Symp. Circuits and Systems, Newton, 1975.

[9] J. Bruan, "Topological analysis of networks containing nullators and norators", Electr. Letters, p. 427, 1966.

[10] F. Brtkovic and M. Vehovec, "Matrix reduction method for solution of sparse circuit equations", Proc. IEEE Int. Symp. Circuits and Systems, pp. 175-179, 1978.

[11] J.R. Bunch and D.J. Rose, "Partitioning, tearing and modification of sparse linear systems", J. Math. Anal. Appl., vol. 48, pp. 574-593, Nov. 1974.

[12] E.M. Butler, E. Cohen, N.J. Elias, J.J. Golembeski and R.G. Olsen, "Capitol-Circuit Analysis program Including Tolerances", Proceedings ISCAS, pp. 570-574, Phoenix, 1977.

[13] H.J. Carlin, "Singular network elements", IEEE Trans. Circuit Theory, vol. CT-11, pp. 67-72, March 1964.

[14] 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, pp. 358-361, The Hague, 1981.

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

[16] S.P. Chan and S.G. Chan, "Modification of toplogical formulas", IEEE Trans. Circuit Theory, vol. CT-15, pp. 84-86, March 1968.

[17] J.P. Char, "Generation of trees, two-trees and storage of master forests", IEEE Trans. Circuit Theory, vol. CT-15, pp. 228-238, Sept. 1968.

[18] C.T. Chen, Introduction to linear systems theory, New York: Holt, Rinehard and Winston, 1970.

[19] W.K. Chen, Applied graph theory - graphs and electrical networks, Amsterdam: North-Holland, 1976.

[20] W.K. Chen, "Flow graphs: some properties and methods of simplifications", IEEE Trans. Circuit Theory, vol. CT-12, pp. 128-130, 1965.

[21] W.K. Chen, "Topological analysis for active networks", IEEE Trans. Circuit Theory, vol. CT-12, pp. 85-91, March 1965.

[22] 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.

[23] W.K. Chen and H.C. Li, "Computer generation of directed trees and complete trees", Int. J. Electronics, vol. 34, pp. 1-21, 1973.

[24] L.O. Chua and L.K. Chen, "Diakoptic and generalized hybrid analysis", IEEE Trans. Circuits and Systems, vol. CAS-23, pp. 694-705, Dec. 1976.

[25] L.O. Chua and L.K. Chen, "On optimally sparse cycle and coboundary basis for a linear graph", IEEE Trans. Circuit Theory, vol. CT-20, pp. 495-503, Sept. 1973.

[26] L.O. Chua and P.M. Lin, Computer aided analysis of electronic circuits-algorithms and computational techniques, Englewood Cliffs, N.J.: Prentice-Hall, 1975.

[27] A. Cichocki and S. Osowski, "Matrix Coates flow-graphs", IEE Journal on Electric Circuits and Systems, vol. 2, pp. 199-204, 1978.

[28] J.O. McClenhanan and S.P. Chan, "Computer analysis of general linear networks using digraphs", Int. Journal of Electronics, vol. 33, pp. 153-191, 1972.

[29] C.L. Coates, "Flow graph solutions of linear algebraic equations", IRE Trans. Circuit Theory, vol. CT-6, pp. 170-187, 1959.

[30] C.L. Coates, "General topological formulas for linear networks functions", IRE Trans. Circuit Theory, vol. 5, pp. 42-54, March 1958.

[31] J. Cajka, "New formula for the signal-flow graph reduction", Electr. Letters, pp. 437-438, July 1970.

[32] A.C. Davies, "Nullator-norator equivalent networks for controlled sources", Proc. IEEE, pp. 722-723, 1967.

[33] A.C. Davies, "On matrix analysis of networks containing nullators and norators", Electronic Letters, p. 48, 1966.

[34] A.C. Davies, "The significance of nullators, norators and nullors in active network theory", Radio Electron. Eng., vol. 34, pp. 259-267, 1967.

[35] A.C. Davies, "Topological solutions of networks containing nullators and norators", Electr. Letters, p. 90, 1966.

[36] N. Deo, Graph theory with applications to engineering and computer sciences, Englewood Cliffs, NJ: Prentice-Hall, 1974.

[37] R.J. Duffin, "An analysis of Wang algebra of network", Trans. Am. Math. Soc., Oct. 1959.

[38] W.R. Dunn and S.P. Chan, "Analysis of active networks by a subgraph-constuction technique", Second Asilomar Conf. Circuits and Systems, 1968.

[39] W.R. Dunn, Jr. and S.P. Chan, "Topological formulation of active network functions", IEEE Trans. Circuit Theory, vol. CT-18, pp. 554-557, Sept. 1971.

[40] A.E. Engel and D.A. Mlynski, "Cliques and partition of graphs", Proc. IEEE Int. Symp. Circuits and Systems, vol. CAS-25, pp. 81-82, 1978.

[41] J.K. Fidler and J.I. Sewell, "Symbolic analysis for computer-aided design - the interpolative approach", IEEE Trans. Circuit Theory, vol. CT-20, pp. 738-741, Nov. 1973.

[42] B.R.M. Gandhi, V.P. Rao and G.S. Raju, "Passive and active circuit analysis by the method of structural numbers", Int. J. Electr., vol. 32, 1972.

[43] N.E. Gibbs, "A cycle generation algorithm for finite undirected linear graphs", Operat. Res., vol. 17, pp. 838-847, 1969.

[44] L.L. Gorsztein, "O razbijenii grafow", Izw. AN SSSR - Techniczeskaja Kibernetika, no. 1, pp. 98-101, 1969.

[45] J.C. Gower, "Comparison of some methods of cluster analysis", Biometrics, vol. 54, pp. 623-637, Dec. 1967.

[46] I.N. Hajj, "Sparsity considerations in network solution by tearing", Proc. IEEE Int. Symp. Circuits and Systems, pp. 170-174, 1978.

[47] S.L. Hakimi and D.G. Green, "Generation and realization of trees and k-trees", IEEE Trans. Circuit Theory, vol. CT-11, pp. 247-255, June 1964.

[48] H.H. Happ, Diakoptics and networks, New York: Academic Press, 1971.

[49] H.H. Happ, "Diakoptics - the solution of system problems by tearing", Proc. IEEE, vol. 62, pp. 930-940.

[50] W.W. Happ, "Flowgraph techniques for closed systems", IEEE Trans. Aerospace and Electronic Systems, vol. AES-2, pp. 252-264, May 1966.

[51] R. Hashemian, "Symbolic representation of network transfer functions using norator-nullator pairs", Electronic Circuits and Systems, vol. 1, pp. 193-197, Nov. 1977.

[52] C.W. Ho, A.E. Ruehli and P.A. Brennan, "The modified nodal approach network analysis", IEEE Trans. Circuits and Systems, vol. CAS-22, pp. 504-509, July 1975.

[53] J.E. Hopcroft and R.E. Tarjan, "Dividing a graph into triconnected components", SIAM J. Comput., vol. 2, no. 3, 1973.

[54] M.T. Jony and G.W. Zobrist, "Topological formulas for general linear networks", IEEE Trans. Circuit Theory, vol. CT-15, pp. 251-259, Sept. 1968.

[55] I. Kaufman, "On poles and zeros of linear systems", IEEE Trans. Circuits and Systems, vol. CAS-20, pp. 93-101, March 1973.

[56] J. Katzenelson and S. Tsur, "On the symbolic analysis of linear networks", IEEE Trans. Circuit Theory, vol. CT-20, pp. 572-574, 1973.

[57] B.W. Kernighas and S. Lin, "An efficient heuristic procedure for partitioning graphs", Bell System Tech. J., vol. 49, pp. 291-307, Feb. 1970.

[58] Sh. N. Khusainov, "Topological formulas for electric circuits with multipole elements", Trans. Izv. Akad. Nauk SSSR Energ. and Transp., vol. 12, pp. 160-164, 1974.

[59] W.H. Kim and R.T. Chien, Topological analysis and synthesis of communication networks, New York: Columbia, University Press, 1962.

[60] G. Kirchhoff, "On the solution of the equations obtained from the investigation of the linear distribution of galvanic currents", IRE Trans. Circuit Theory, CT-5, Mar. 1958.

[61] G. Kishi and Y. Kajitani, "Topological considerations on minimal sets of variables describing an LCR network", IEEE Trans. Circuit Theory, vol. CT-20, pp. 335-340, July 1973.

[62] H. Kitozawa and M. Sagawa, "P-parameter: a new representation of linear network functions and its application to network analysis", Proc. Int. Symp., pp. 1048-1051, Tokyo, 1979.

[63] A. Konczykowska, "Analiza ukladow elektrycznych z zastosowaniem metod dekompozycji", Ph.D. Dissertation, Techn. Univ. of Warsaw, Warszawa, 1976.

[64] A. Konczykowska, "Computer program of analysis of decomposed networks", Proc. European Conf. Circuit Theory and Design, Genova, 1976.

[65] A. Konczykowska and J. Starzyk, "Computer analysis of large signal flowgraphs by hierarchical decomposition method", Proc. European Conf. Circuit Theory and Design, pp. 408-413, Warszawa, 1980.

[66] A. Konczykowska and J. Starzyk, "Computer justification of upward topological analysis of signal-flow graphs", Proc. European Conf. Circuit Theory and Design, pp. 464-467, The Hague 1981.

[67] A. Konczykowska and J. Starzyk, "Hierarchical decomposition of signal-flow graphs", III Int. Conf. on Electronic Circuits, Prague, pp. 155-160, 1979.

[68] A. Konczykowska and J. Starzyk, "Wyznaczanie liczby strukturalnej grafu zdekomponowanego, Czesc I i II", Arch. Elektrot., z. 2, 1975.

[69] G. Kron, "A set of principles to interconnect the solution of physical systems", Journal of Applied Physics, vol. 24, pp. 956-980, Aug. 1953.

[70] G. Kron, Diakoptics: the piecewise solution of large scale systems, London: McDonald, 1963.

[71] G. Kron, Tensor analysis of networks, London: John Wiley and Sons, 1965.

[72] E.S. Kuh and R.A. Rohre, "The state-variable approach to network analysis", Proc. IEEE, vol. 53, July 1965.

[73] P.M. Lin, "A survey of applications of symbolic network functions", IEEE Trans. Circuit Theory, vol. CT-20, pp. 732-737, 1973.

[74] P.M. Lin and G.E. Alderson, "SNAP-A computer program for generating symbolic network functions", School Elec. Eng., Purdue Univ., Lafayette, Ind. Techn. Dep. TR-EE, 70-16, Aug. 1970.

[75] F. Luccio and M. Sami, "On the decomposition of networks in minimally interconnected subnetworks", IEEE Trans. Circuit Theory, vol. CT-16, pp. 184-188, May 1969.

[76] H. Mann and J. Kvasil, "An alorithm for the formulation of state-space equations", Proc. Int. Symp. Circuits and Systems, pp. 161-162, Tokyo, 1979.

[77] S.J. Mason, "Feedback theory - some properties of signal flow graphs", Proc. IRE, 41, pp. 1144-1156, Sept. 1953.

[78] S.J. Mason, "Feedback theory - further properties of signal flow graphs", Proc. IRE, 44, pp. 920-926, July 1956.

[79] S.J. Mason and H. Zimmermann, Electronic circuits, signals and systems, New York: J. Wiley, 1960.

[80] L.M. Maxwell and M.B. Reed, The theory of graphs: a basis for network theory, New York: Pergamon Press, 1971.

[81] W. Mayeda and S. Seshu, "Generation of trees without duplications", IEEE Trans. Circuit Theory, CT-12, pp. 181-185, June 1965.

[82] W.J. McCalla and P.O. Pederson, "Elements of computer-aided analysis", IEEE Trans. Circuit Theory, vol. CT-18, pp. 14-26, Jan. 1971.

[83] M.D. McIlroy, "Algorithm 354: Generator of spanning trees", Comm. ACM, vol. 12, p. 511, Sept. 1969.

[84] O.P. McNamee and N. Potash, "A user's and programmer's manual for NASAP", Univ. California, Los Angeles, Rep. 63-68, Aug. 1968.

[85] R.R. Mielke, "A new signal flow graph formulation of symbolic network functions", IEEE Trans. Circuits and Systems, vol. CAS-25, pp. 334-340, June 1978.

[86] G.J. Minty, "A simple algorithm for listing all the trees of a graph", IEEE Trans. Circuit Theory, vol. CT-12, p. 120, 1965.

[87] J. Mulawka, "Biwezlowe grafy Masona i ich zastosowanie do analizy i syntezy ukladow RC ze wzmacniaczami operacyjnymi", Ph.D. dissertation, Technical Univ. of Warsaw, Warszawa, 1976.

[88] J. Nadratowski, "Analiza liniowych ukladow elektroniczynch w oparciu o metody topologiczne i technike maszynowego liczenia", Ph.D. Dissertation, Technical Univ. of Warsaw, Warszawa, 1971.

[89] A. Nathan, "Topological rules for linear networks", IEEE Trans. Circuit Theory, vol. CT-12, pp. 344-358, Sept. 1965.

[90] A.R. Newton, "Techniques for the simulation of large-scale integrated circuits", IEEE Trans. Circuits and Systems, vol. CAS-26, pp. 741-749, Sept. 1979.

[91] T. Nitta and A. Kishima, "State equation of linear networks with dependent sources based on network topology", Trans. Electr. Commun., Japan, vol. I60-A, pp. 1046-1053, 1977.

[92] J. Numata and M. Iri, "Mixed-type topological formulas for general linear networks", IEEE Trans. Circuit Theory, vol. CT-20, pp. 488-494, Sept. 1973.

[93] C.E. Ogbuobiri, W.F. Tinney and J.W. Walker, "Sparsity-directed decomposition for Gaussian elimination on matrices", IEEE Trans. Power, Appr. Syst., vol. PAS-89, pp. 141-150, Jan. 1970.

[94] U. Ozguner, "Signal flow graph analysis using only loops", Electr. Letters, pp. 359-360, Aug. 1973.

[95] M. Piekarski, "Listing of all possible trees of a linear graph", IEEE Trans. Circuit Theory, vol. CT-12, p. 124-125, March 1965.

[96] N.B. Rabat and H.Y. Hsieh, "A latent macromodular approach to large-scale networks", IEEE Trans. Circuits and Systems, vol. CAS-23, pp. 745-752, Dec. 1976.

[97] D.E. Riegle and P.M. Lin, "Matrix signal flow graphs and an optimum topological method for evaluating their gains", IEEE Trans. Circuit Theory, vol. CT-19, pp. 427-435, 1972.

[98] S.M. Roberts and B. Flores, "Systematic generation of Hamiltomian circuits", Comm. ACM, vol. 9, pp. 690-694, 1966.

[99] I.W. Sandberg and H.S. So, "A two-sets of eigenvalues approach to the computer analysis of linear systems", IEEE Trans. Circuit Theory, vol. CT-16, pp. 509-517, 1969.

[100] A. Sangiovanni-Vincentelli, L.K. Chen and L.O. Chua, "An efficient heuristic cluster algorithm for tearing large-scale networks", IEEE Trans. Circuits and Systems, vol. CAS-24, pp. 709-717, Dec. 1977.

[101] A. Sangiovanni-Vincentelli, L.K. Chen and L.O. Chua, "Node tearing nodal analysis", Electronics Research Laboratory, University of California, Berkeley, Memo ERL-M582, Sept. 1976.

[102] P. Sannuti and N.N. Puri, "Generation of symbolic network functions via exterior algebra", Proc. Int. Symp. Circuits and Systems, pp. 190-194, New York, 1978.

[103] P. Sannuti and N.N. Puri, "Symbolic network analysis - and algebraic formulation", Proc. IEEE Int. Symp. Circuits and Systems, pp. 1044-1047, Tokyo, 1979.

[104] S. Seshu and M.B. Reed, Linear graphs and electrical networks, Reading, Mass: Addison-Wesley, 1961.

[105] S.D. Shieu and S.P. Chan, "Topological formulas of symbolic network functions for linear active networks", Proc. Int. Symp. Circuit Theory, pp. 95-98, 1973.

[106] K. Singhal and J. Vlach, "Generation of immitance functions in symbolic form for lumped-distributed active networks", IEEE Trans. Circuits and Systems, vol. CAS-21, pp. 57-67, Jan. 1974.

[107] K. Sinhal and J. Vlach, "Symbolic analysis of analog and digital circuits", IEEE Trans. Circuits and Systems, vol. CAS-24, pp. 598-609, Nov. 1977.

[108] E.V. Sorensen, "A linear semisymbolic circuit analysis program based on algebraic eigenvalue technique", Report. Techn. University of Denmark, Lyngbey 1972.

[109] J.A. Starzyk, "An efficient cluster algorithm", Proc. 5th Chech-Polish Workshop Circuit Theory, Podiebrady, 1980.

[110] J.A. Starzyk and E. SLiwa, "Hierarchic decomposition method for the topological analysis of electronic networks", Int. J. Circuit Theory and Applications, vol. 8, pp. 407-417, 1980.

[111] J.A. Starzyk, "Signal flow-graph analysis by decomposition method", IEE Proc. Electronic Circuits and Systems, no. 2, pp. 81-86, 1980.

[112] J.A. Starzyk, "Synteza topologiczna liniwych ukladow stacjonarnych metoda liczb strukturalnych", Ph.D. Dissertation, Technical Univ. of Warsaw, Warszawa, 1976.

[113] J.A. Starzyk, "The dispersor graphs", Proc. of 3rd Czech-Polish Workshop Circuit Theory, Prenet 1978.

[114] J.A. Starzyk and E. Sliwa, "Topological analysis by hierarchic decomposition method", IV Int. Symp. Network Theory, pp. 155-160, Ljubljana, 1979.

[115] J.A. Starzyk, "Analiza topologiczna duzych ukladow elektronicznych", Prace Naukowe, Elektronika, No. 55, WPW, Warszawa, 1981.

[116] E. Sliwa, "Hierarchiczna analiza topologiczna i jej zastosowanie do analizy i projektowania ukladow elektronicznych", Ph.D. Dissertation, Technical University of Warsaw, Warszawa, 1982.

[117] A. Talbot, "Topological analysis for active networks", IEEE Trans. Circuit Theory, vol. CT-13, pp. 111-112, March 1966.

[118] J.C. Tiernan, "An efficient search algorithm to find the elementary circuits of a graph", Comm. ACM, vol. 13, pp. 722-726, Dec. 1970.

[119] S.L. Ting, "On the general properties of electrical network determinant", Chinese J. Phys., no. 1, 1935.

[120] C.T. Tsai, "Short cut methods for expanding the determinants involved in network problems", Chinese J. Phys., No. 3, 1939.

[121] M.K. Tsai, B.A. Shenoi, "Generation of symbolic network functions using computer software techniques", IEEE Trans. Circuits and Systems, vol. CAS-26, pp. 344-346, June 1979.

[122] L. Weinberg, Network analysis and synthesis, New York: McGraw-Hill, 1962.

[123] J. Wojciechowski, "Grafy blokowe i ich zastosowanie w analizie liniwoych ukladow elektrycznych", Ph.D. Dissertation, Technical University of Warsaw, Warszawa, 1976.

[124] W.T. Wang, "On a new method for the analysis of electrical networks", Nat. Res. Inst. for Engineering, Academia Sinica Memoir, no. 2, 1934.

[125] H.A. Weinblatt, "New search algorithm for finding the simple cycles of a finite directed graph", J. ACM, no. 1, pp. 43-56, 1972.

[126] F. Wu, "Solution of large-scale networks by tearing", IEEE Trans. Circuits and Systems, vol. CAS-23, pp. 706-713, Dec. 1976.

[127] Algorithm 459, The elementary circuits of a graph, Comm. ACM, 16, pp. 632-633, 1973.

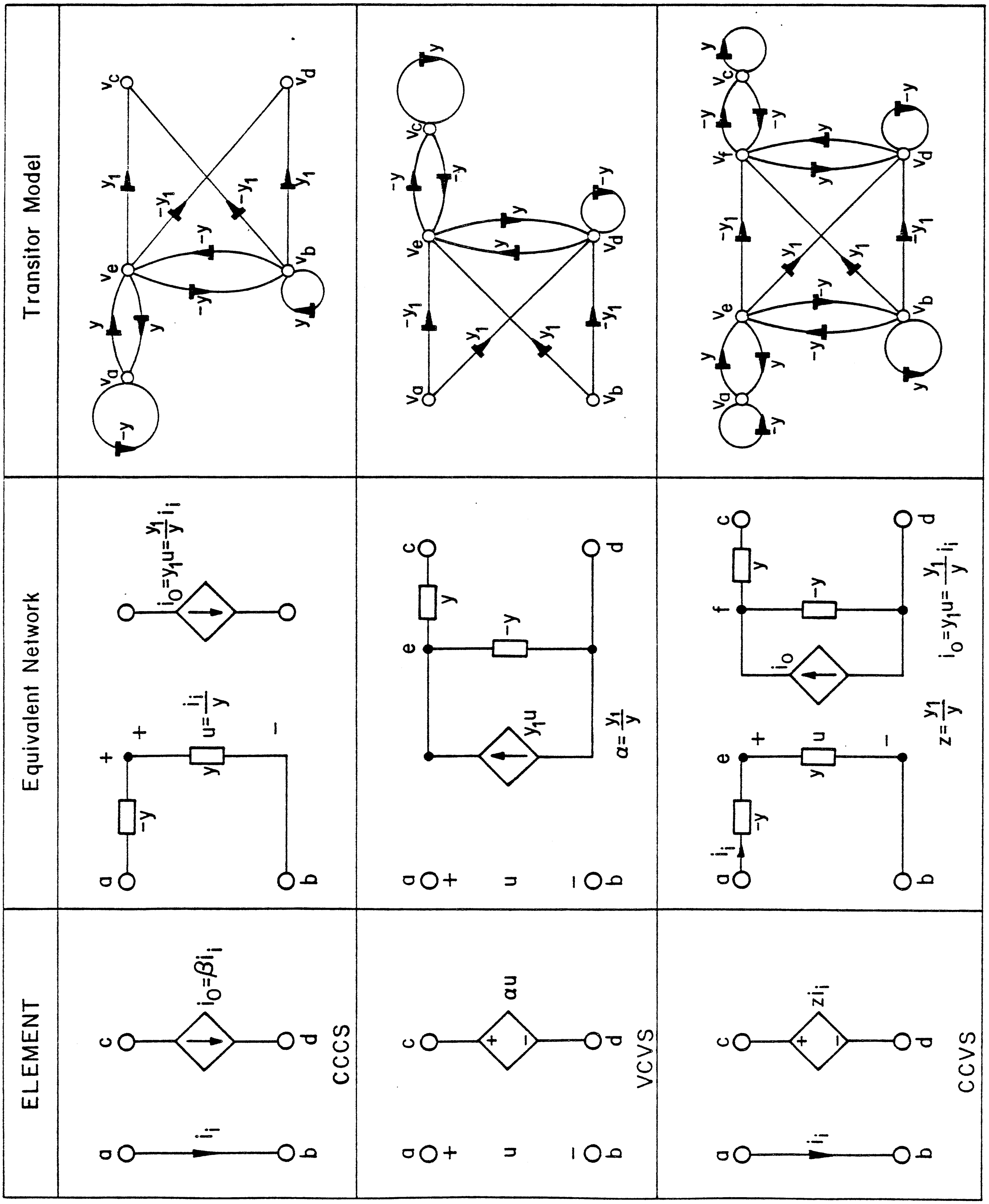


TABLE I

TRANSITOR MODELS OF CONTROLLED SOURCES

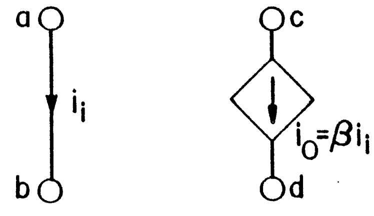
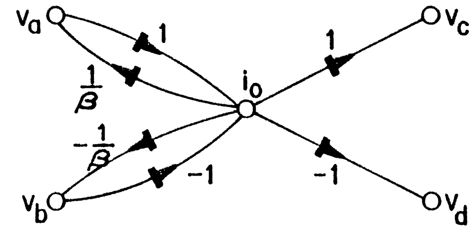
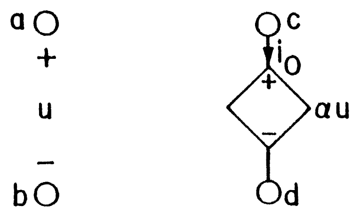
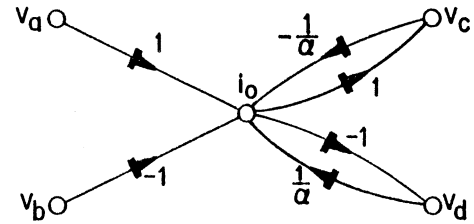
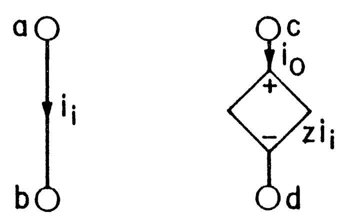
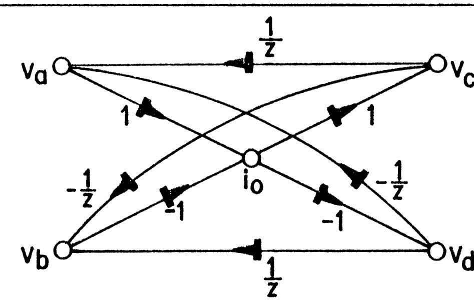
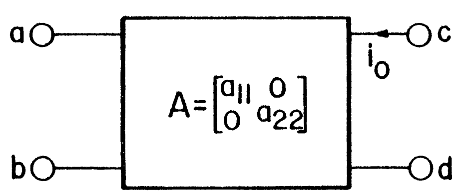
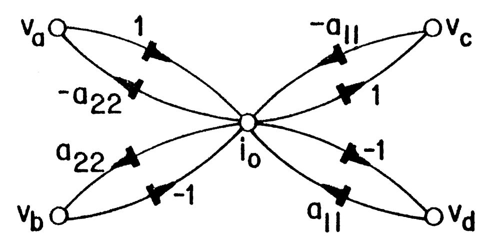
ELEMENT	GRAPH	EQUATIONS
 CCCS		$v_a - v_b = 0$ $i_a = \frac{1}{\beta} i_o$, $i_b = -\frac{1}{\beta} i_o$ $i_c = i_o$, $i_d = -i_o$
 VCVS		$v_a - v_b - \frac{1}{\alpha}(v_c - v_d) = 0$ $i_a = 0$, $i_b = 0$ $i_c = i_o$, $i_d = -i_o$
 CCVS		$v_a - v_b = 0$ $i_a = \frac{1}{z}(v_c - v_d)$, $i_b = \frac{1}{z}(v_d - v_c)$ $i_c = i_o$, $i_d = -i_o$
 Converter		$v_a - v_b - a_{11}(v_c - v_d) = 0$ $i_a = -a_{22} i_o$, $i_b = a_{22} i_o$ $i_c = i_o$, $i_d = -i_o$

TABLE II

FORMAL TRANSITOR MODELS

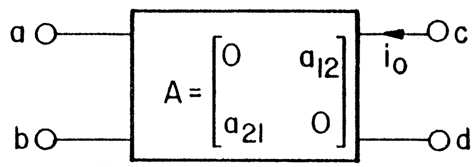
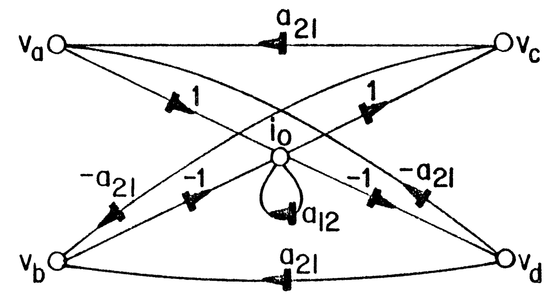
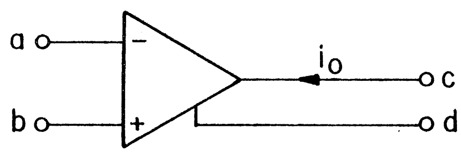
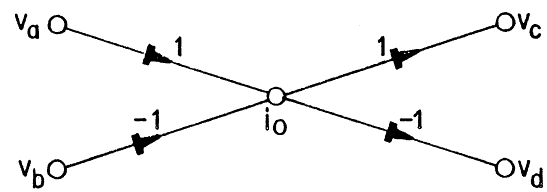
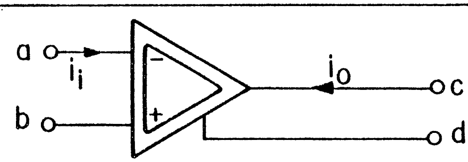
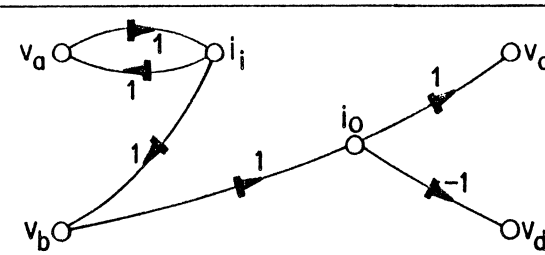
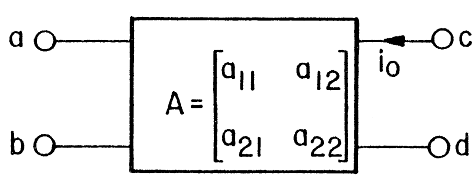
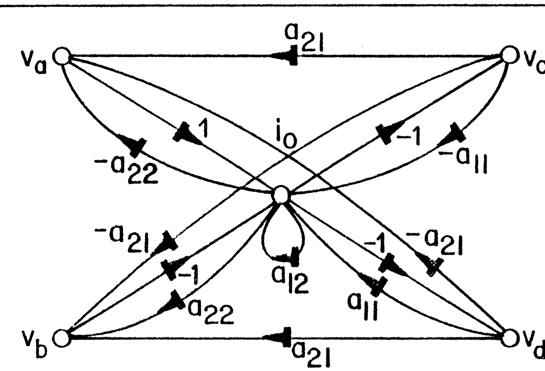
ELEMENT	GRAPH	EQUATIONS
 INVERTER		$v_a - v_b + a_{12} i_o = 0$ $i_a = a_{21}(v_c - v_d)$ $i_b = a_{21}(v_d - v_c)$ $i_c = i_o \qquad i_d = -i_o$
 OP-AMP		$v_a - v_b = 0$ $i_a = 0 \qquad i_b = 0$ $i_c = i_o \qquad i_d = -i_o$
 NORTON AMP.		$v_a = 0 \qquad v_b = 0$ $i_a = i_i \qquad i_b = i_i$ $i_c = i_o \qquad i_d = -i_o$
 Two-Port, chain matrix description		$v_a - v_b - a_{11}(v_c - v_d) + a_{12} i_o = 0$ $i_a = a_{21}(v_c - v_d) - a_{22} i_o$ $i_b = a_{21}(v_d - v_c) + a_{22} i_o$ $i_c = i_o \qquad i_d = -i_o$

TABLE II (continued)

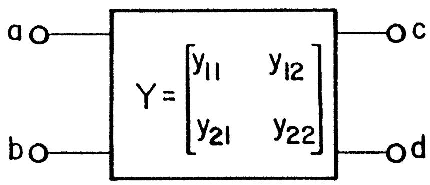
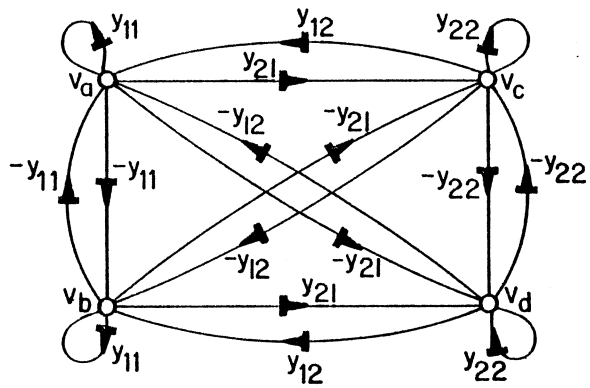
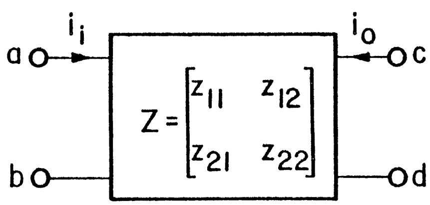
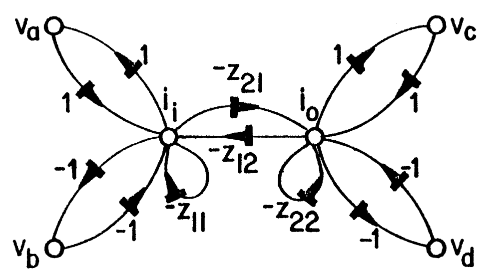
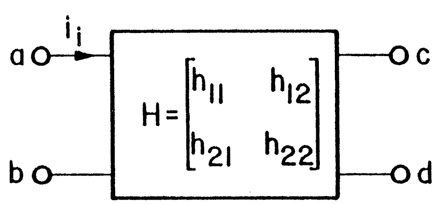
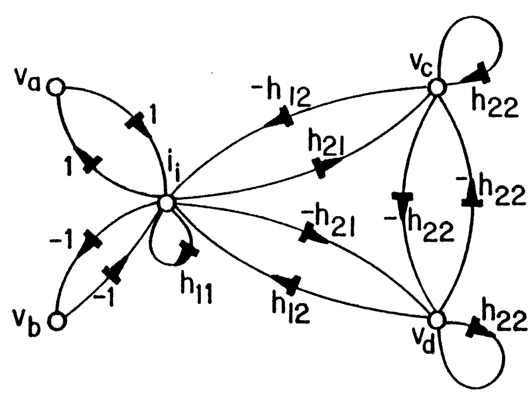
ELEMENT	GRAPH	EQUATIONS
 TWO-PORT admittance description		$i_a = y_{11}(v_a - v_b) + y_{12}(v_c - v_d)$ $i_b = y_{11}(v_b - v_a) + y_{12}(v_d - v_c)$ $i_c = y_{21}(v_a - v_b) + y_{22}(v_c - v_d)$ $i_d = y_{21}(v_b - v_a) + y_{22}(v_d - v_c)$
 Impedance description		$v_a - v_b - z_{11} i_i - z_{12} i_o = 0$ $v_c - v_d - z_{21} i_i - z_{22} i_o = 0$ $i_a = i_i$, $i_b = -i_i$ $i_c = i_o$, $i_d = -i_o$
 Hybrid description		$v_a - v_b - h_{12}(v_c - v_d) - h_{11} i_i = 0$ $i_c = h_{22}(v_c - v_d) + h_{21} i_i$ $i_d = h_{22}(v_d - v_c) - h_{21} i_i$ $i_a = i_i$, $i_b = -i_i$

TABLE II (continued)

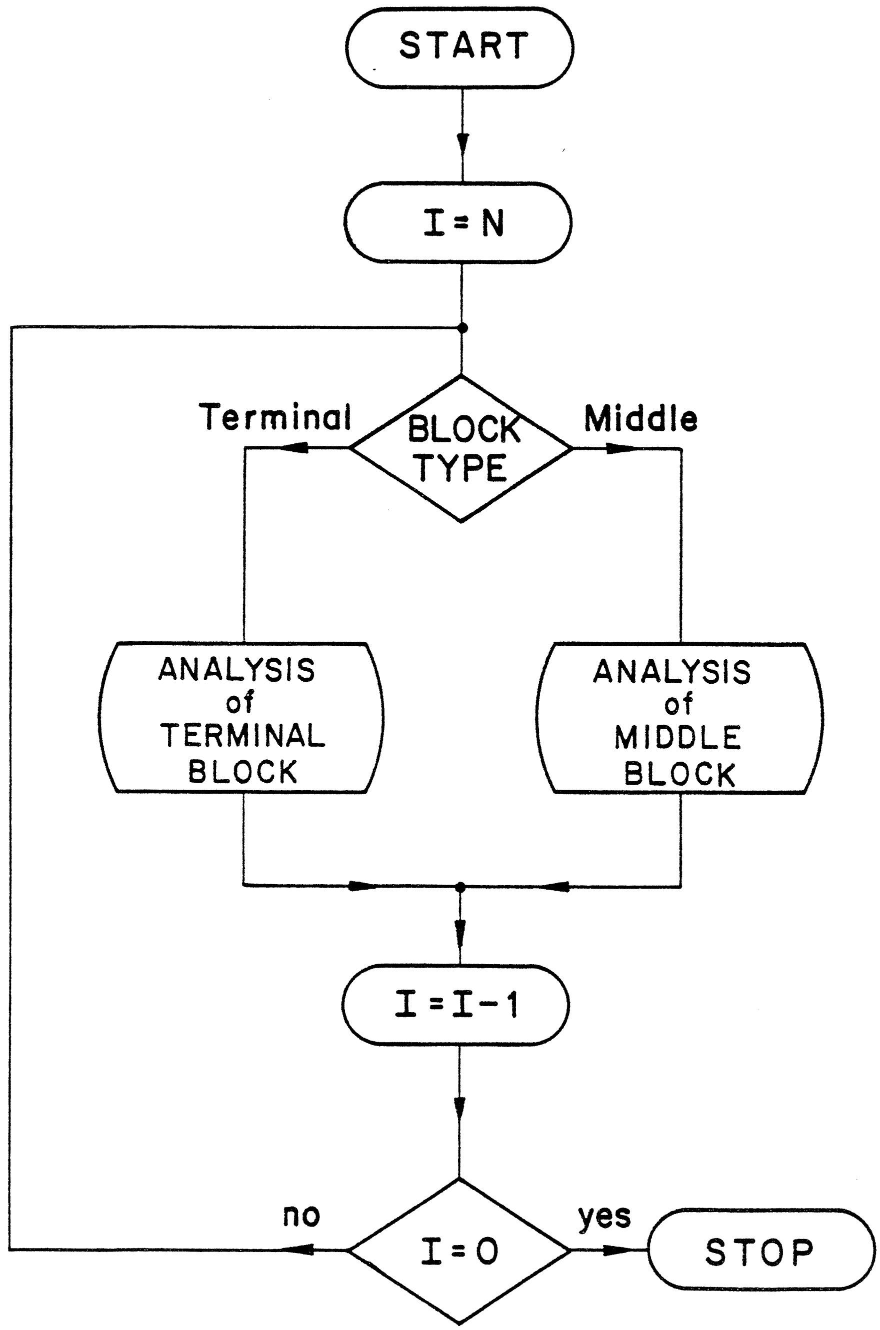


TABLE III

ORGANIZATION OF ALGORITHM OF HIERARCHICAL ANALYSIS

TABLE IV

RESULTS OF SYMBOLIC ANALYSIS

Characteristic Function		Block Number
F(1)	$G_{37}(G_{38} + G_{40})$	9
F(2)	$G_{40}(G_{37} + G_{39})$	
F(3)	$G_{29}G_{32}sC_{35}(G_{28} + G_{30})$	
F(4)	$(G_{28} + G_{30})[G_{32}G_{34}G_{36} + G_{31}sC_{35}(G_{33}+sC_{33})]$	8
F(5)	$G_{28}G_{32}sC_{35}(G_{29} + G_{31} + G_{36})$	
F(6)	$G_{20}G_{23}sC_{26}(G_{19} + G_{21})$	
F(7)	$(G_{19} + G_{21})[G_{23}G_{25}G_{27} + G_{22}sC_{26}(G_{24} + sC_{24})]$	7
F(8)	$G_{19}G_{23}sC_{26}(G_{20} + G_{22} + G_{27})$	
F(9)	$G_{11}G_{14}sC_{17}(G_{16} + G_{12})$	
F(10)	$(G_{10} + G_{12})[G_{14}G_{16}G_{18} + G_{13}sC_{17}(G_{15} + sC_{15})]$	6
F(11)	$G_{10}G_{14}sC_{17}(G_{11} + G_{13} + G_{18})$	
F(12)	$G_2G_5sC_8(G_1 + G_3)$	
F(13)	$(G_1 + G_3)[G_5G_7G_9 + G_4sC_8(G_6 + sC_6)]$	5
F(14)	$G_1G_5sC_8(G_2 + G_4 + G_9)$	

TABLE IV (continued)

F(15)	-F(3)F(1) + F(4)F(2)	4
F(16)	-F(5)F(1)	
F(17)	F(5)F(2)	
F(18)	F(8)F(16)	3
F(19)	F(7)F(15) - F(6)F(17)	
F(20)	-F(8)F(15)	
F(21)	F(11)F(18)	2
F(22)	F(10)F(19) - F(9)F(20)	
F(23)	-F(11)F(19)	
F(24)	F(14)F(21)	1
F(25)	F(13)F(22) - F(12)F(23)	

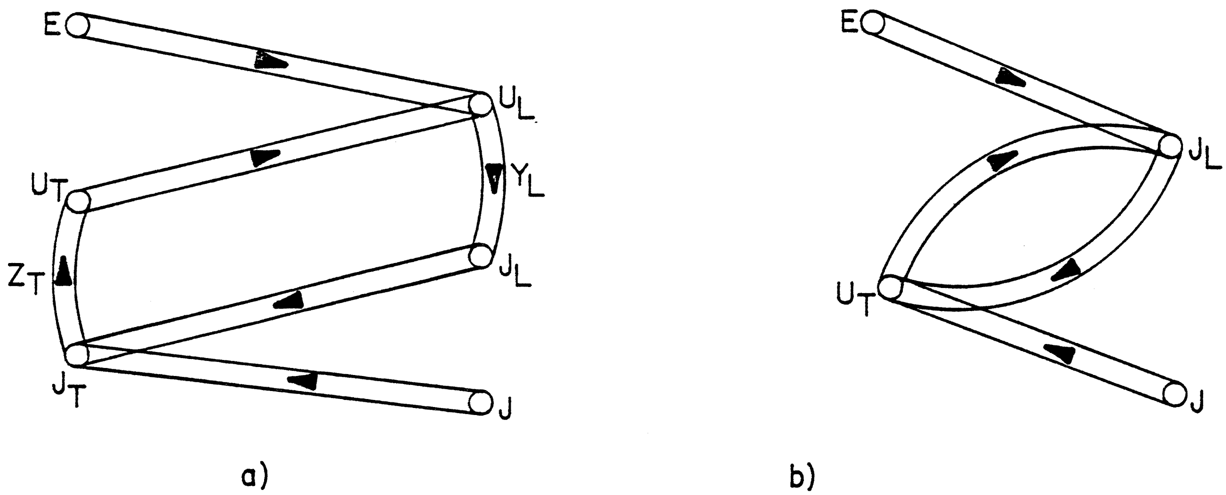


Fig. 1 Matrix graphs for (a) primitive, (b) compact Mason flowgraph.

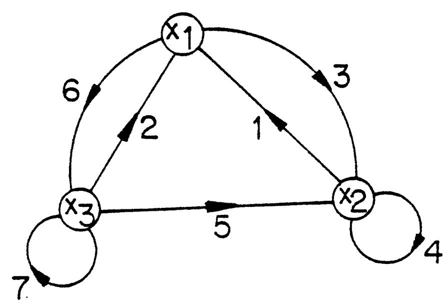


Fig. 2 Coates graph associated with the system of equations (7).

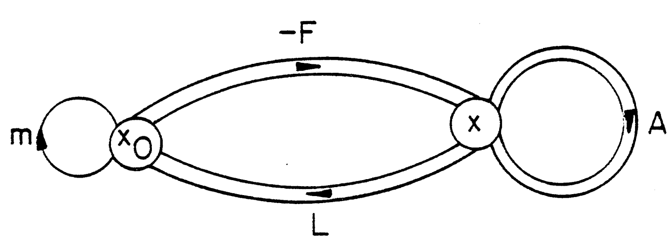


Fig. 3 Matrix Coates graph of extended system of equations (9).

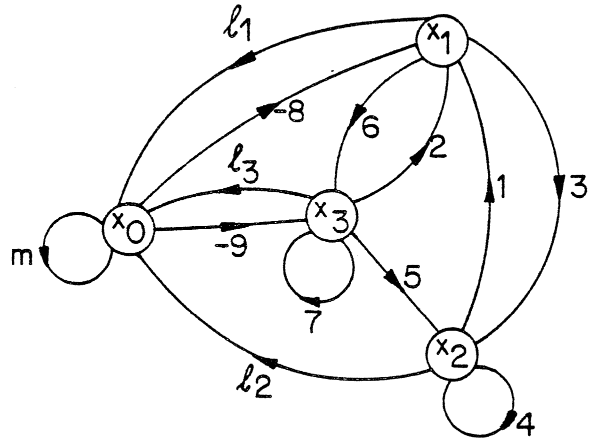


Fig. 4 Coates graph of extended system of equations.

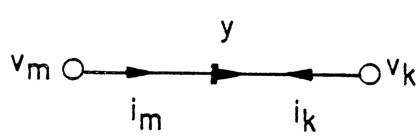


Fig. 5 Symbol for transitor.

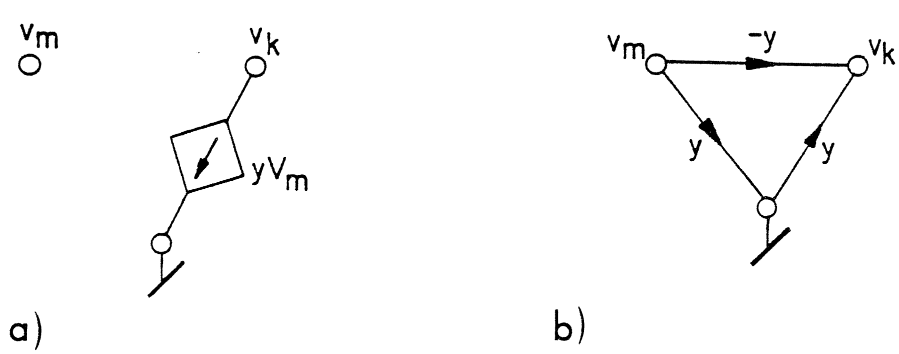


Fig. 6(a) Model of transitor, (b) Equivalent unistor model.

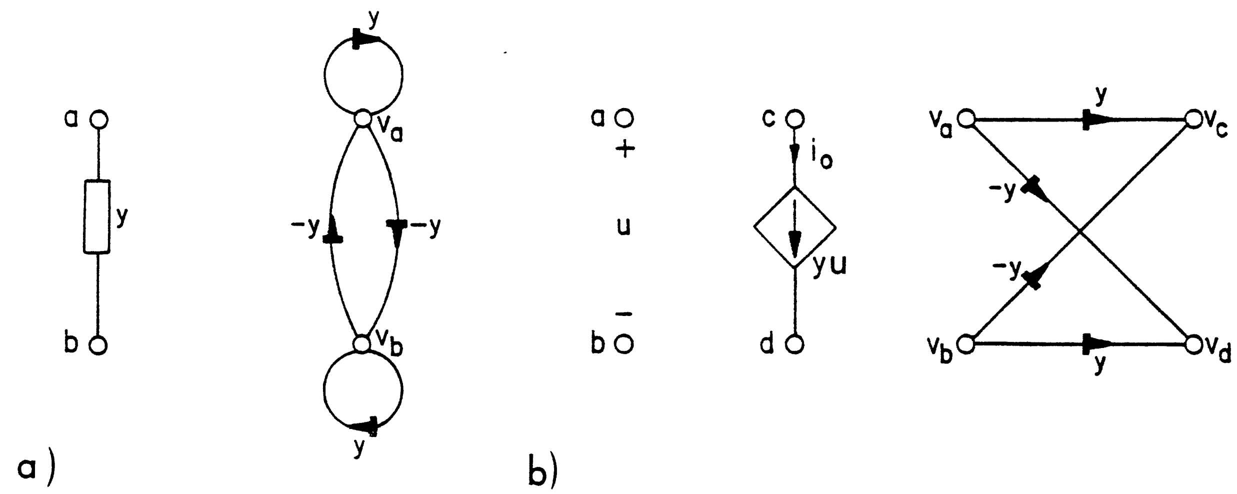


Fig. 7(a) Transitor model of two-terminal, (b) Transitor model of VCCS.

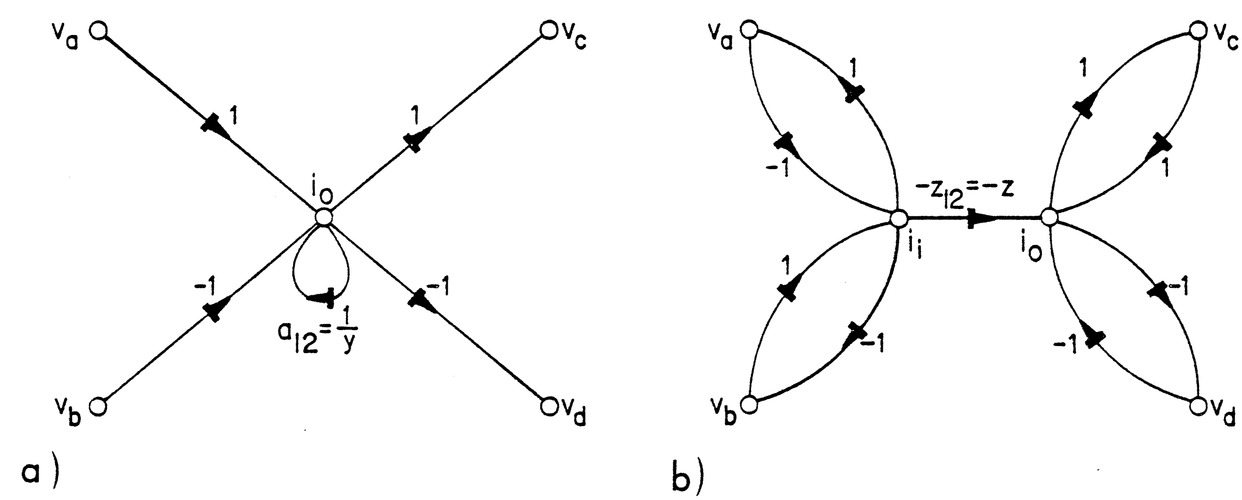


Fig. 8 Formal transitor models (a) VCCS, (b) CCVS.

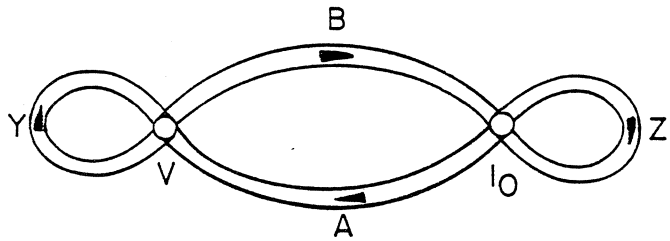


Fig. 9 Matrix Coates graph of multiterminal described by (15).

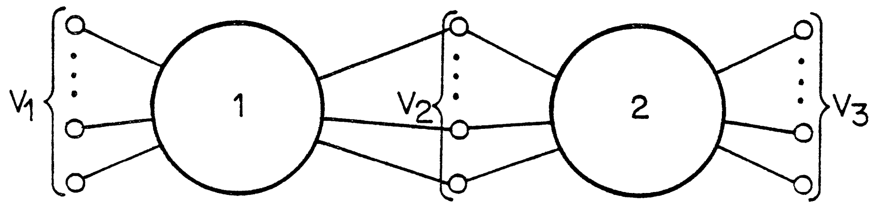


Fig. 10 Connection of two multiterminals.

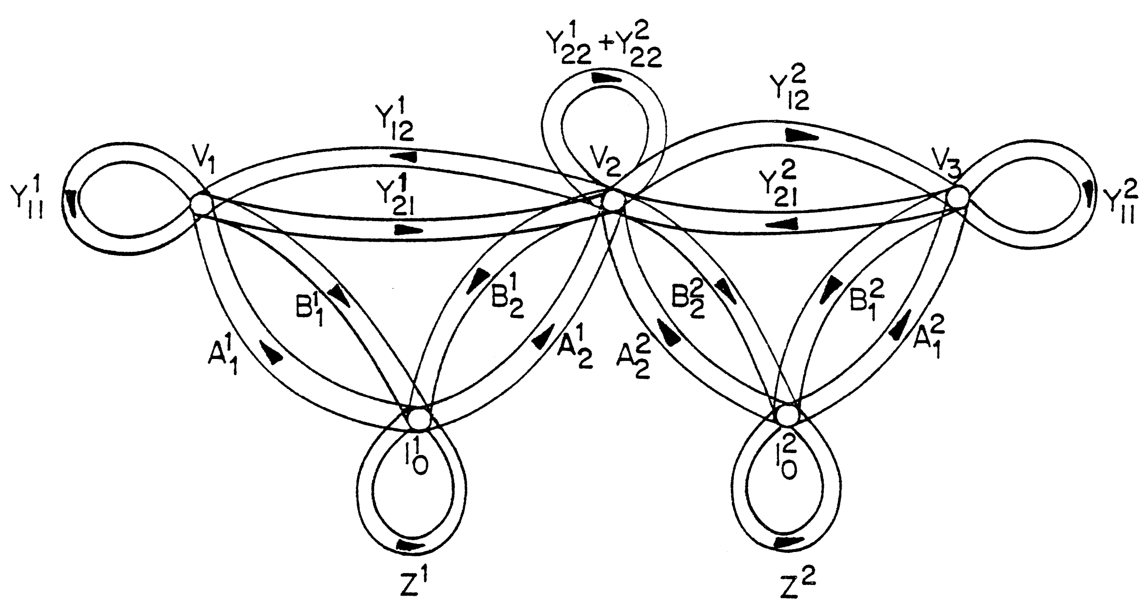


Fig. 11 Matrix Coates graph of the multi-terminal connection.

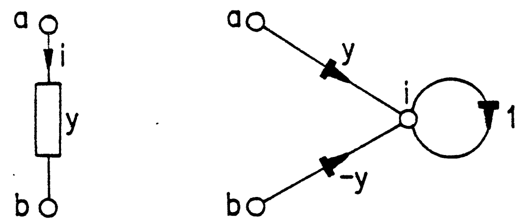


Fig. 12 Consideration of a current in an admittance.

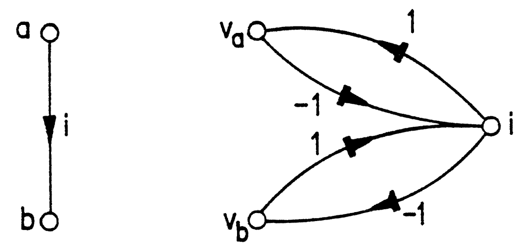


Fig. 13 Consideration of a current in a short circuit.

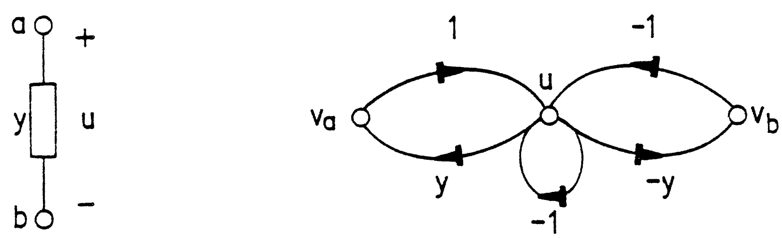


Fig. 14 Formal transitor model of two-terminal.

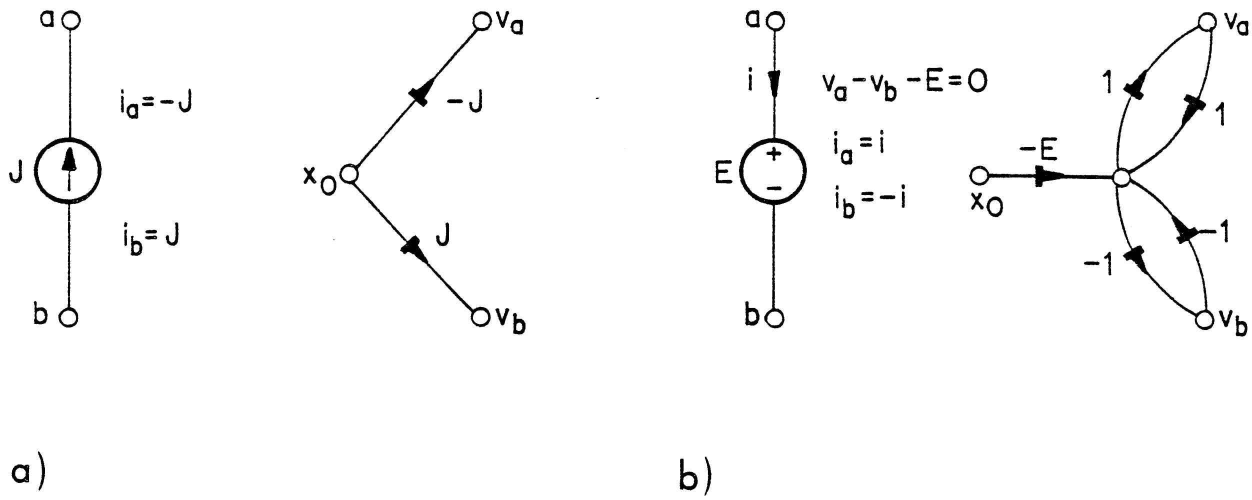


Fig. 15 Transitor models for independent sources (a) current, (b) voltage.

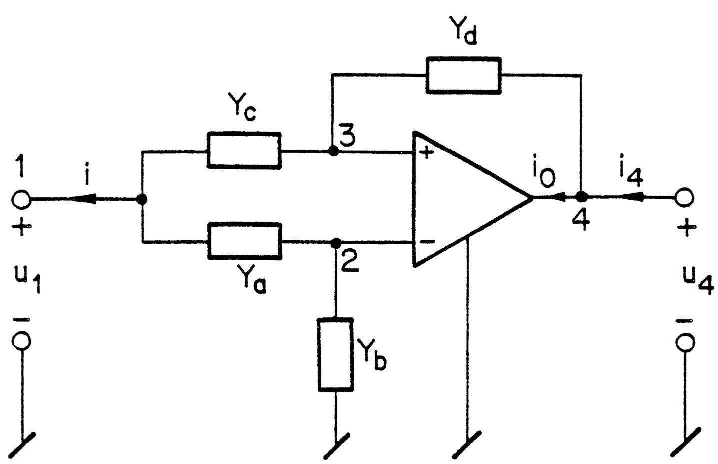


Fig. 16 Circuit with ideal operational amplifier.

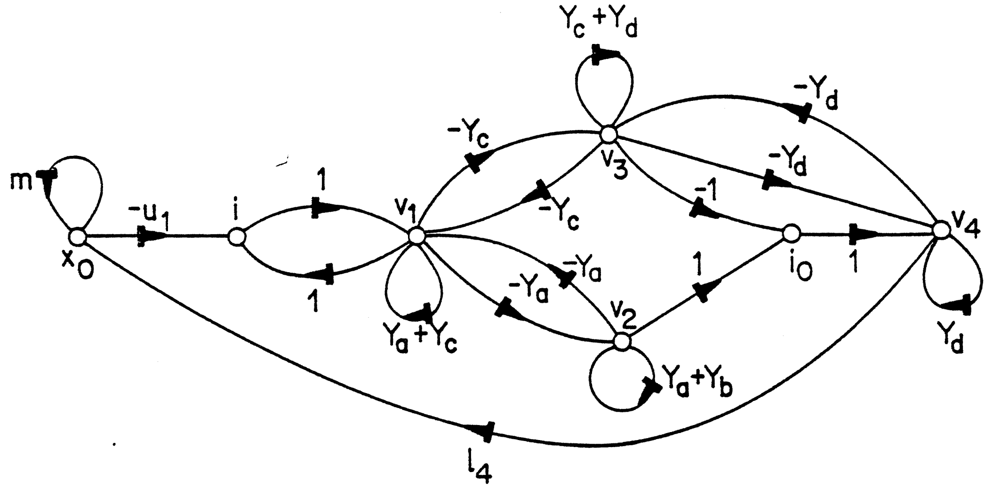


Fig. 17 Transitor graph of the circuit.

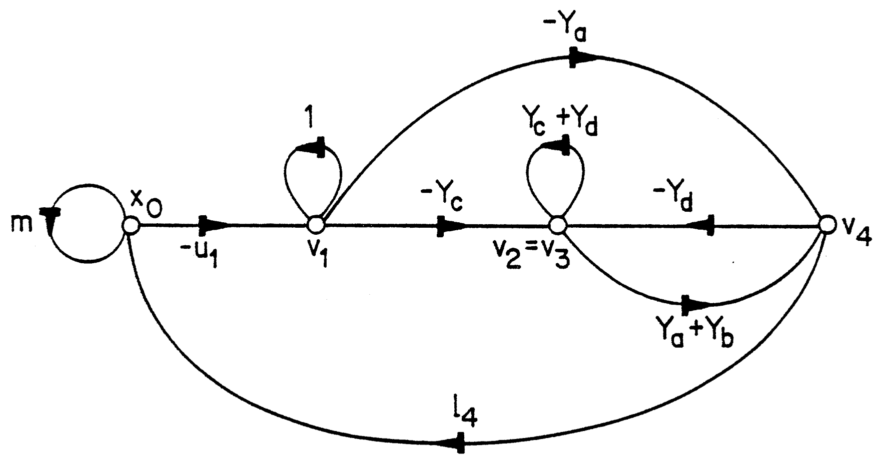


Fig. 18 Transitor graph obtained with an aid of nonautonomous models.

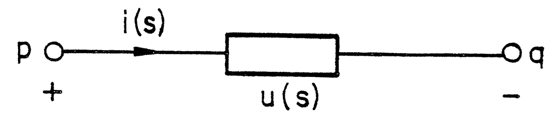


Fig. 19 Two-terminal.

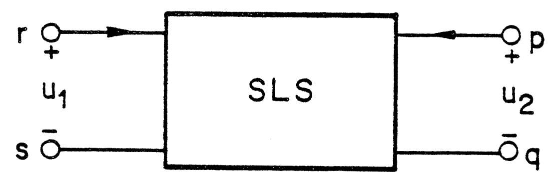


Fig. 20 Two-port.

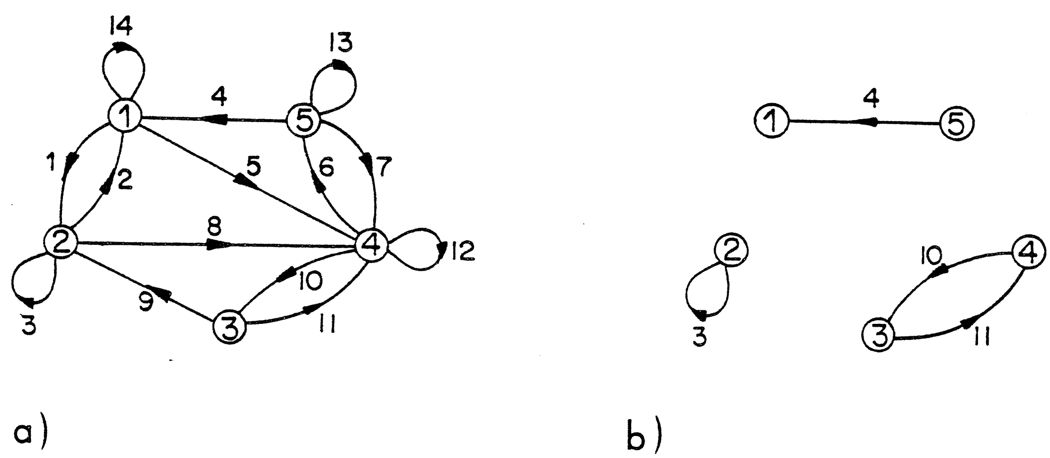


Fig. 21(a) Flowgraph, (b) 1-connection.

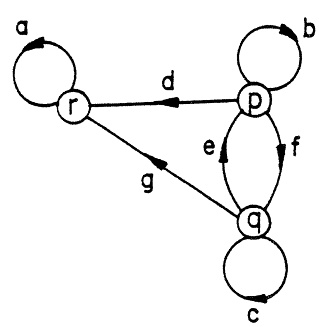


Fig. 22 Connection of auxiliry edges to the flowgraph of two-ports.

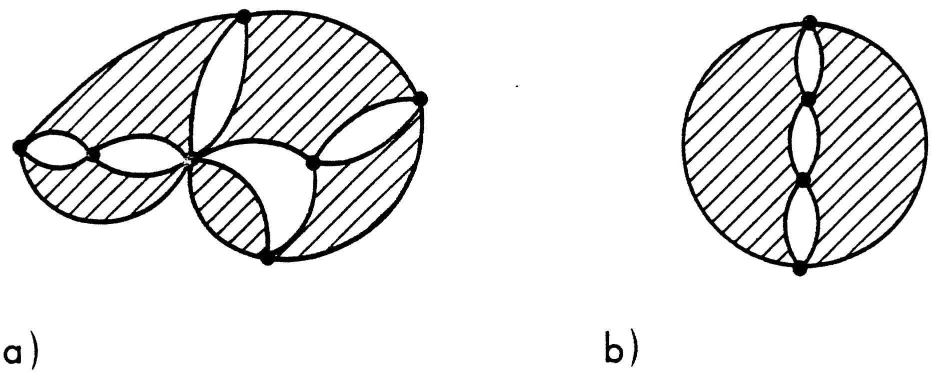


Fig. 23(a) Node decomposition, (b) Fourterminal bisection.

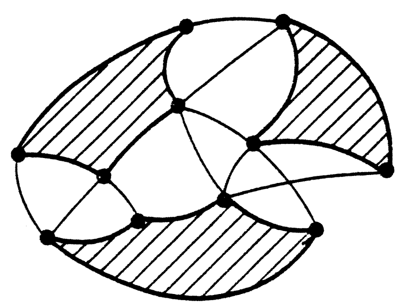


Fig. 24 Edge decomposition.

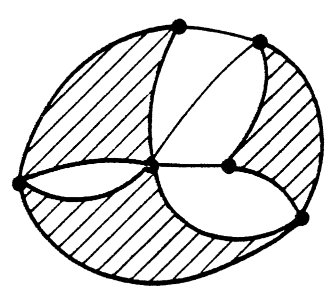


Fig. 25 Hybrid decomposition.

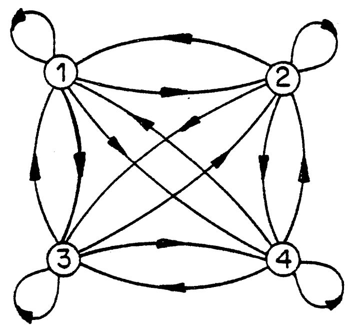


Fig. 26 Substitute graph spanned on 4 block nodes.

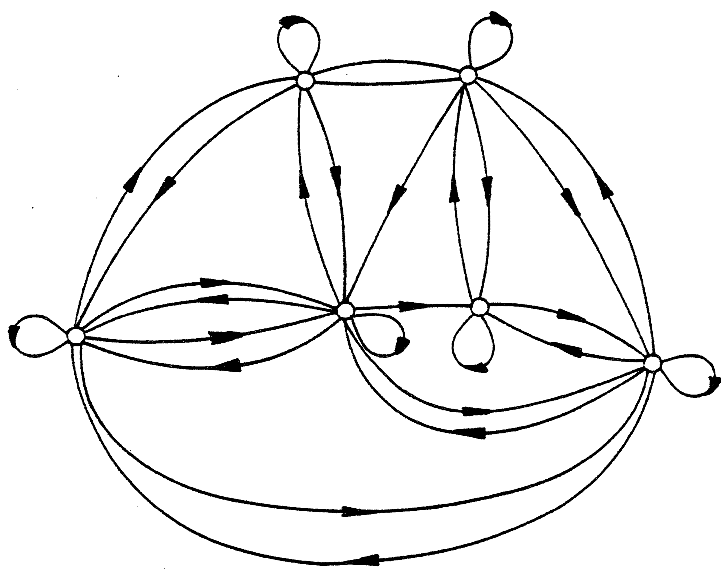


Fig. 27 Decomposition substitute graph.

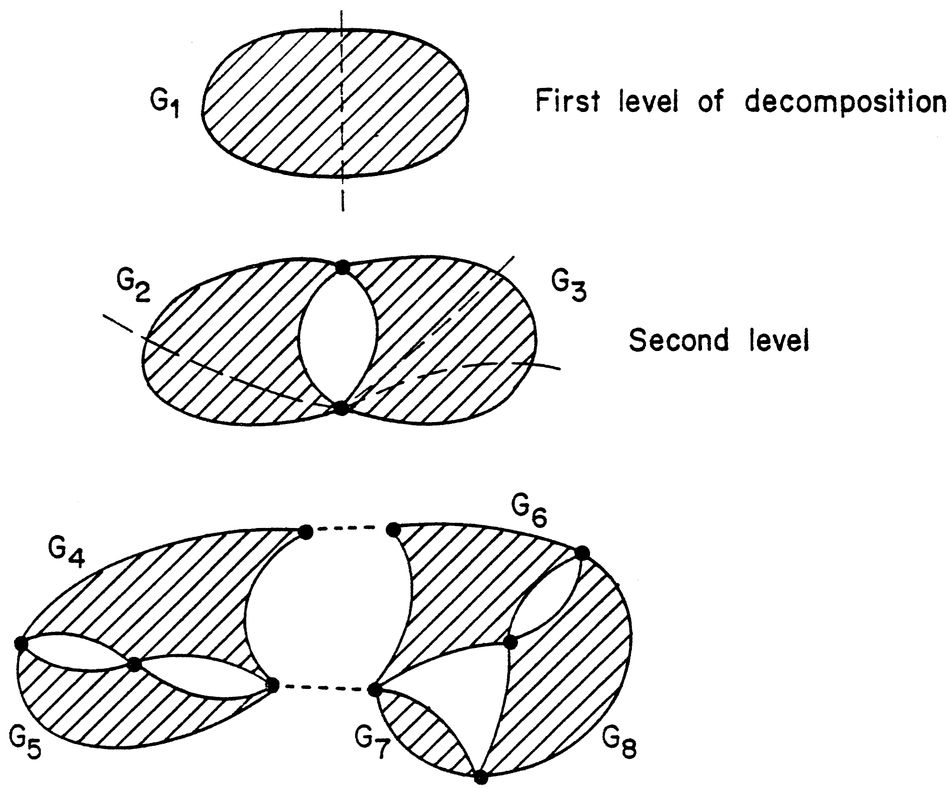


Fig. 28 Two level hierarchical decomposition.

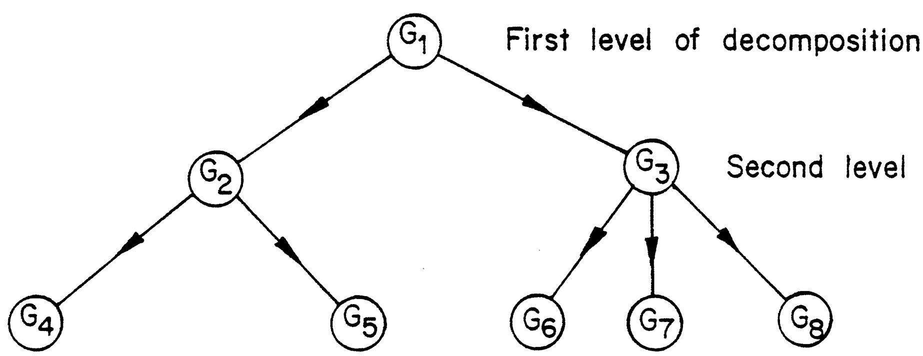


Fig. 29 Tree of decomposition shown in Fig. 28.

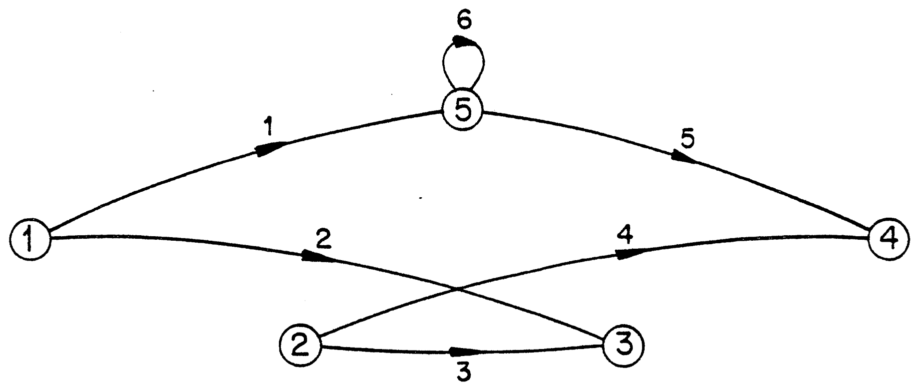


Fig. 30 Proper block.

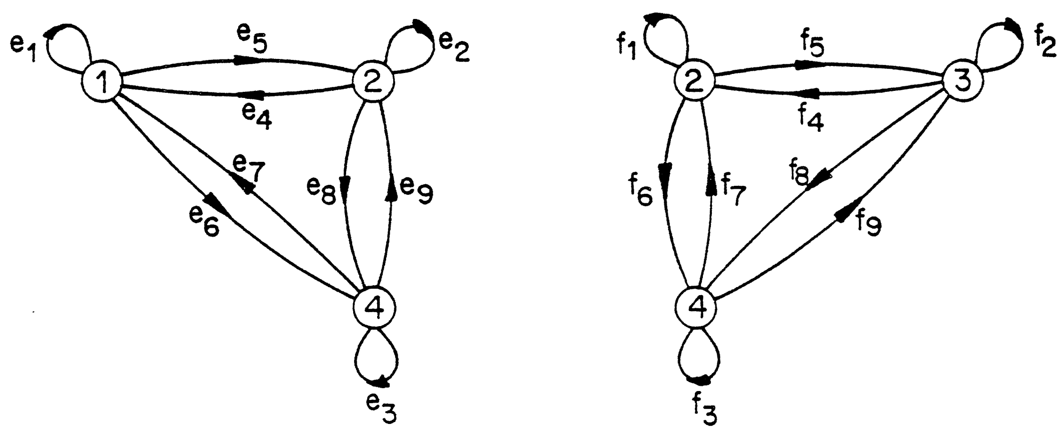


Fig. 31 Blocks to be connected.

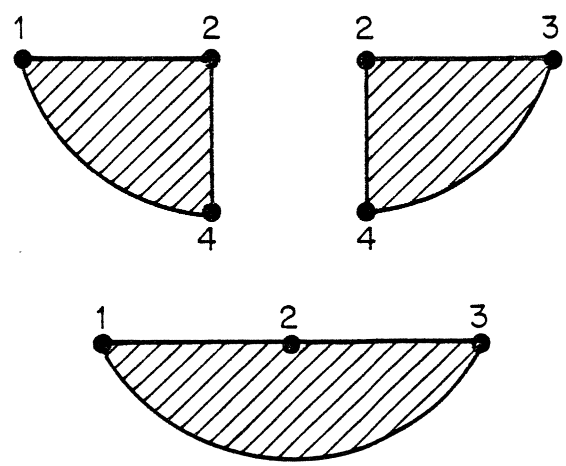


Fig. 32 Association of two blocks.

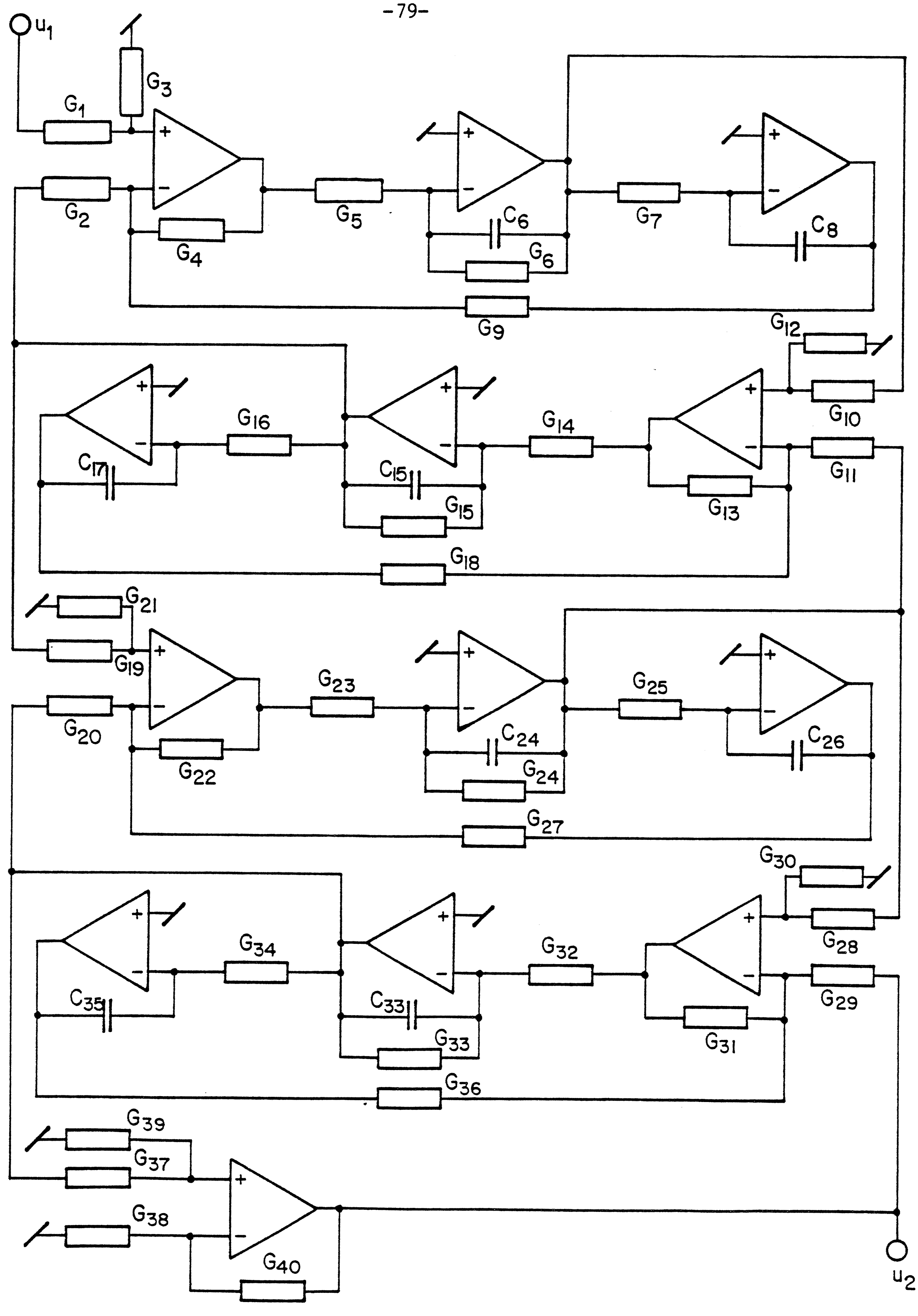


Fig. 33 Band-pass filter.

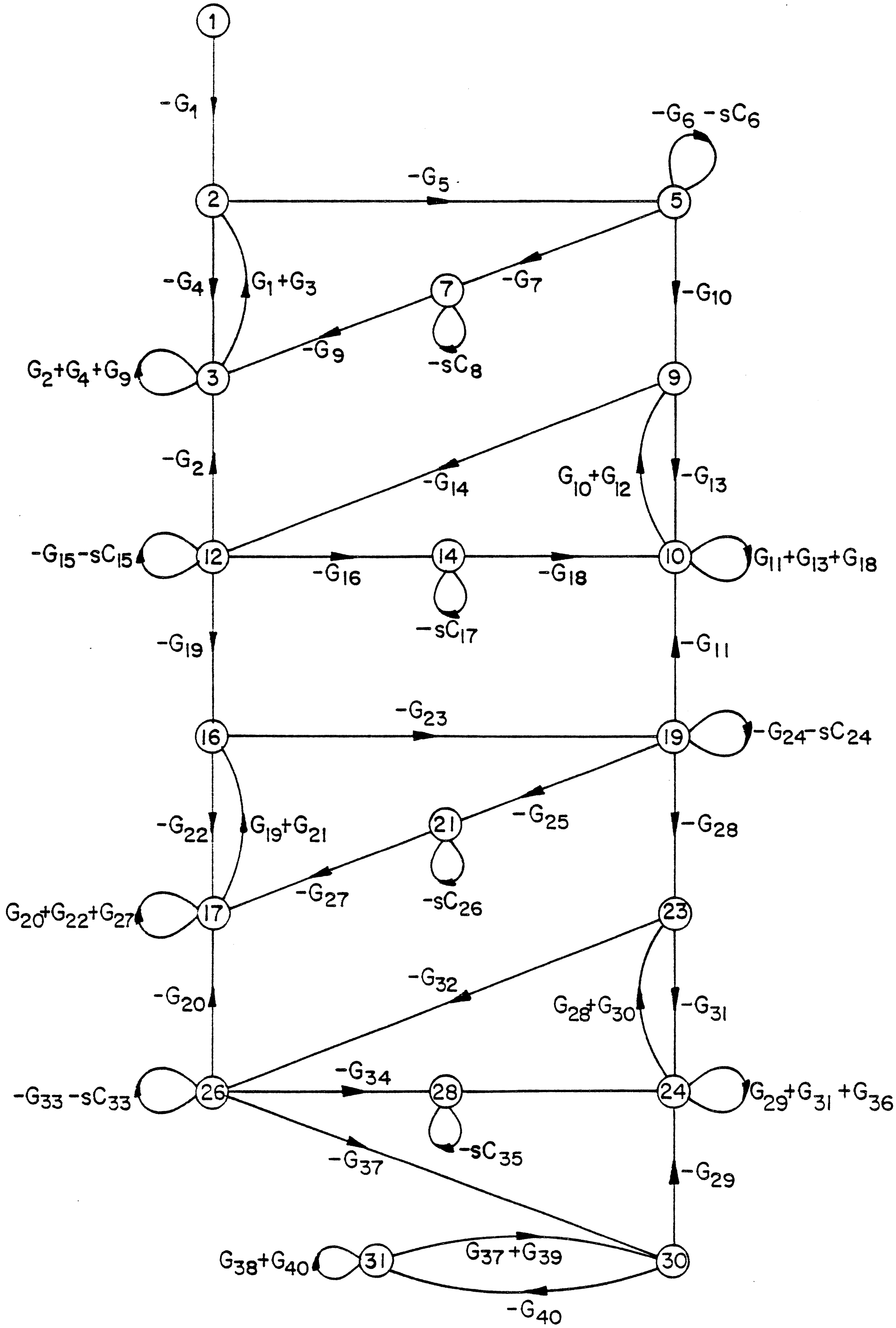


Fig. 34 Flowgraph for band-pass filter.

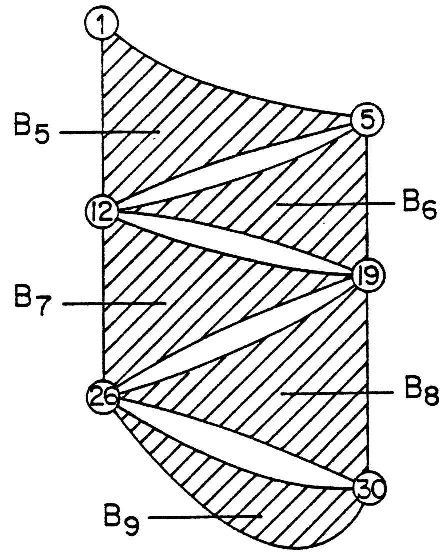


Fig. 35 Block graph.

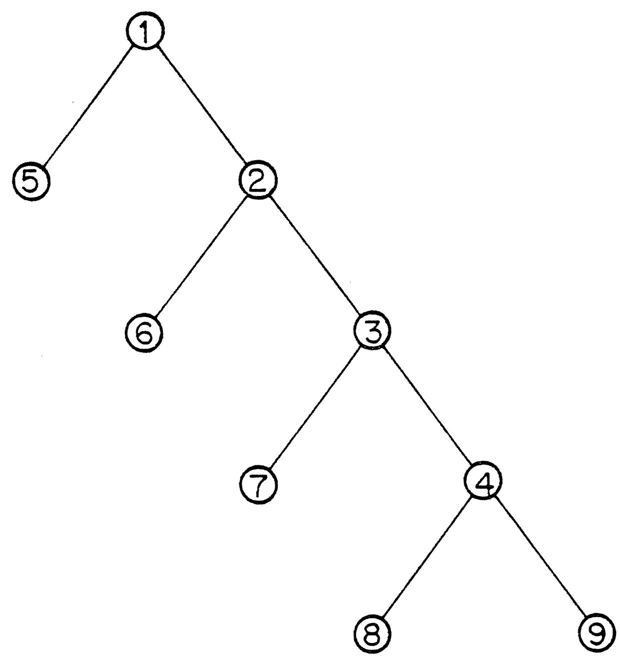


Fig. 36 Tree of hierarchical structure.

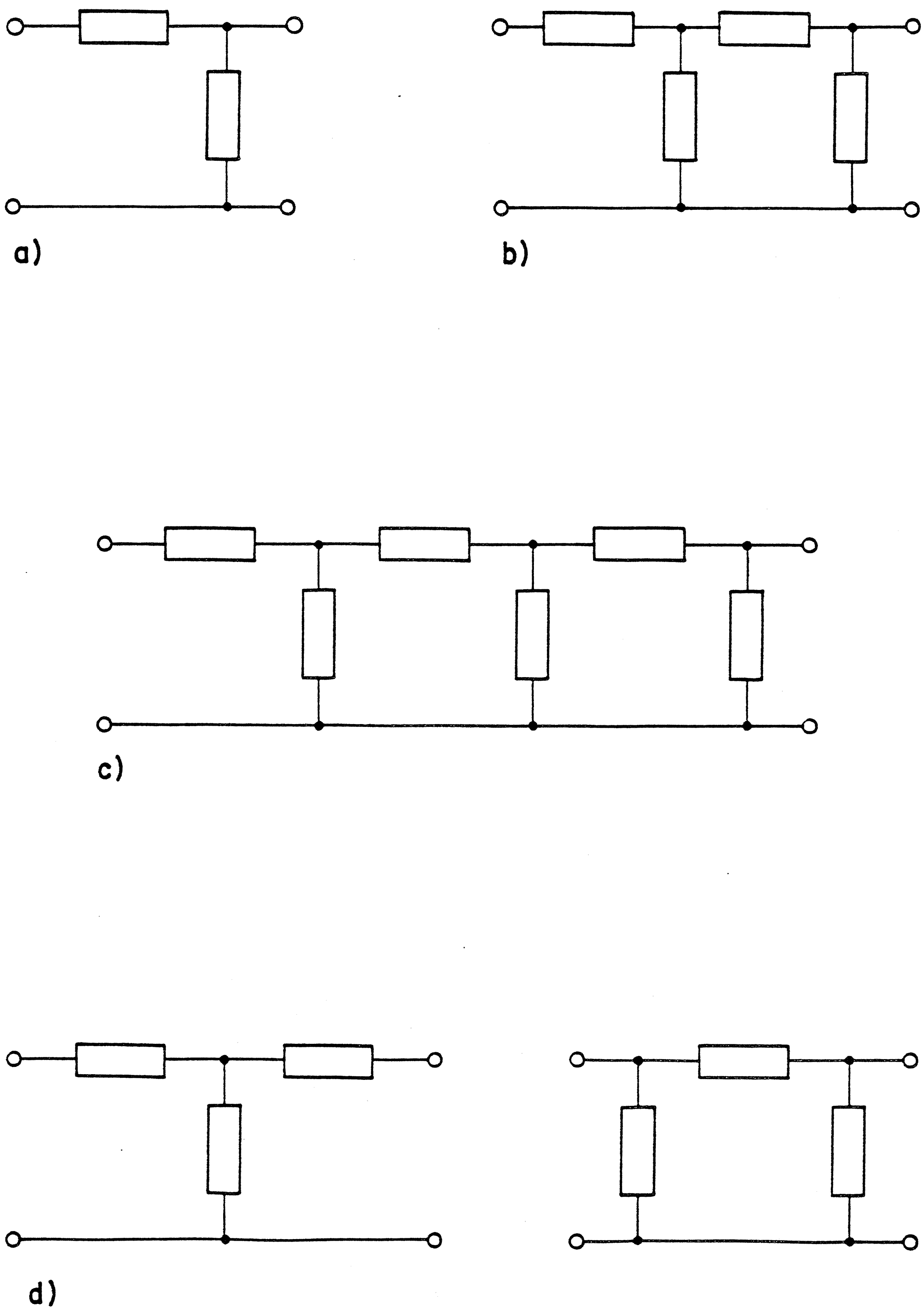


Fig. 37 Terminal blocks of ladder decomposition.

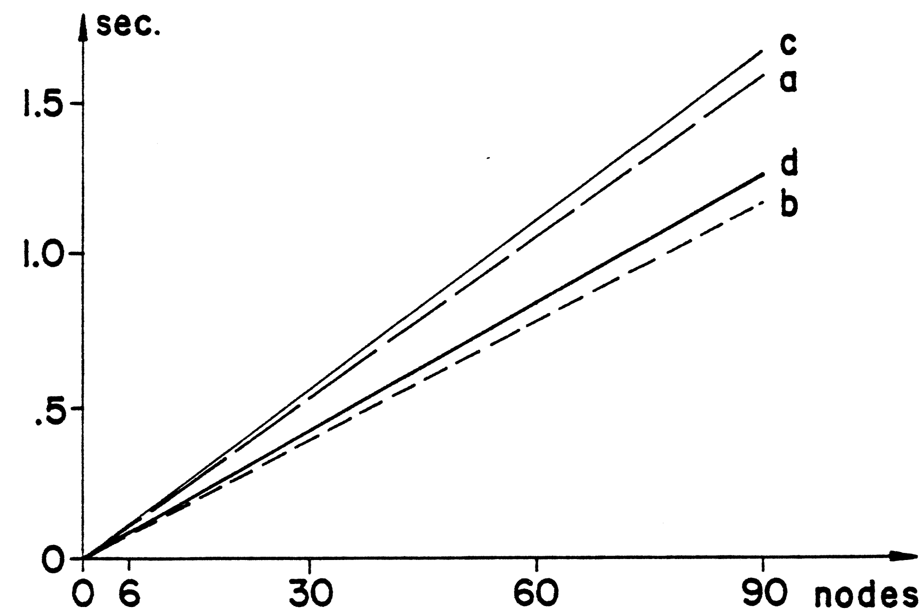


Fig. 38 Relationship between the analysis time and the number of nodes.

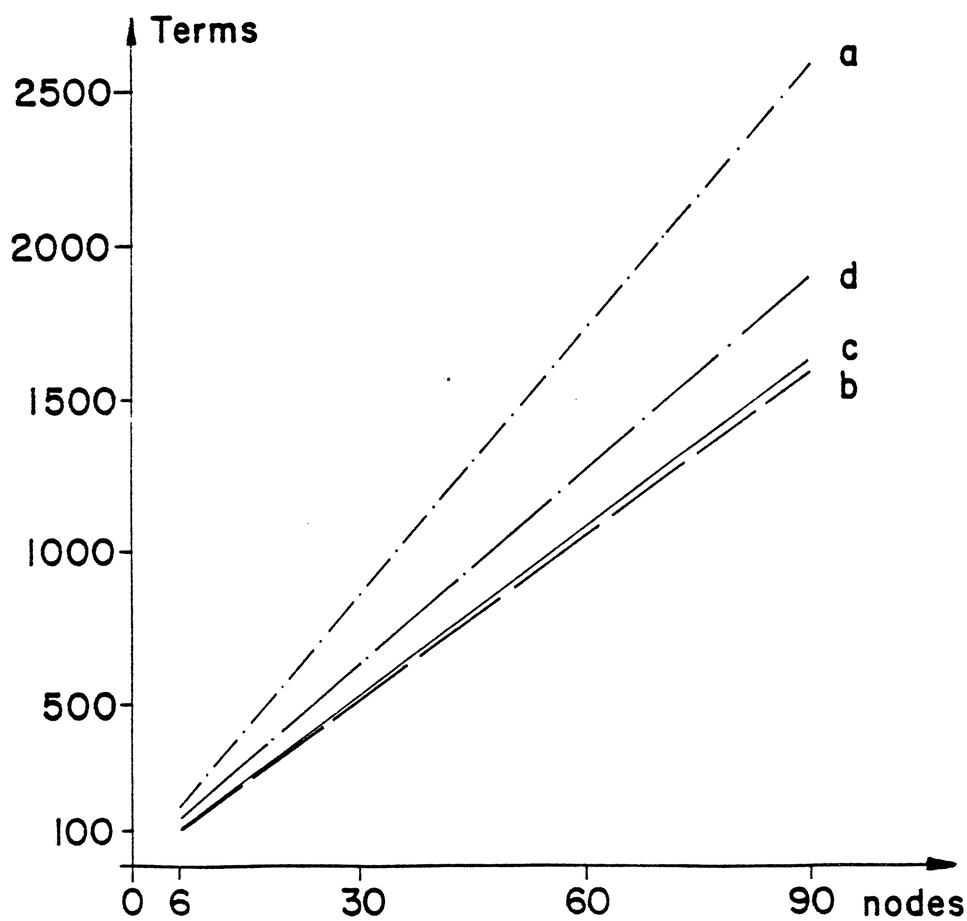


Fig. 39 Relationship between the number of terms and the number of nodes.

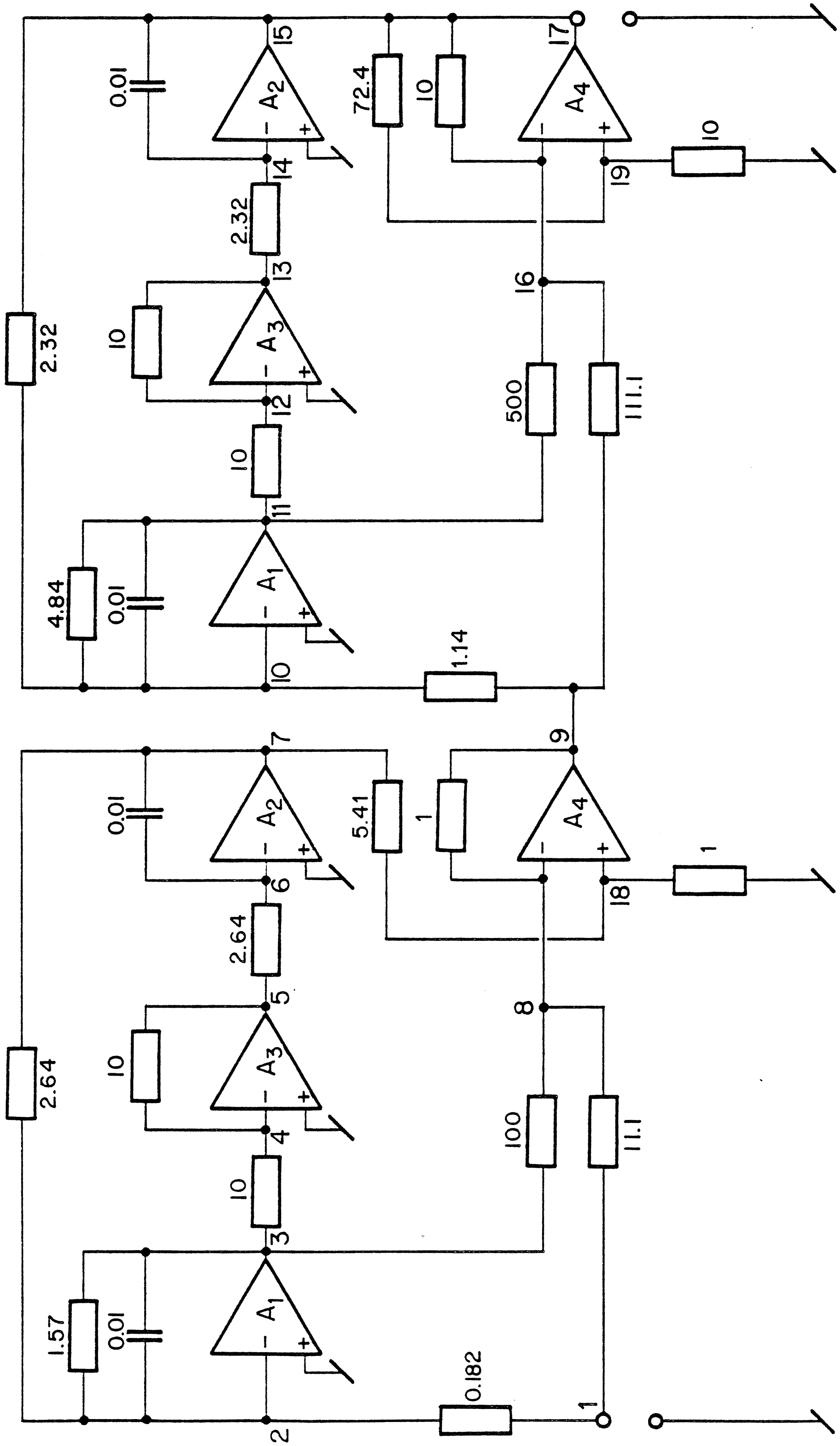


Fig. 40 Lowpass filter network.

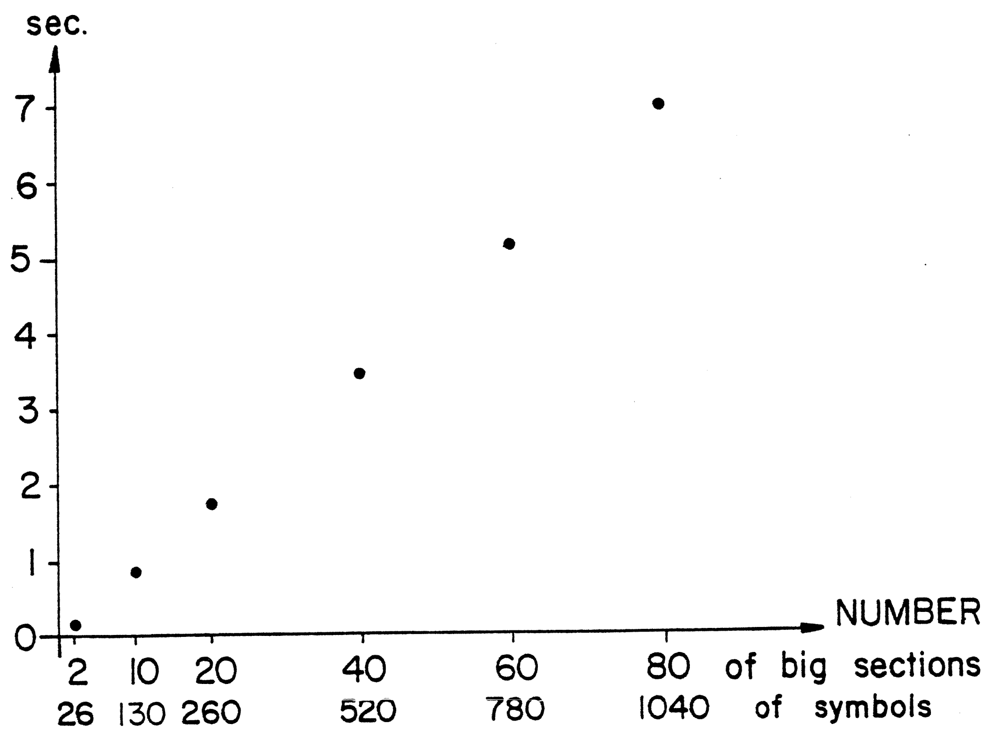


Fig. 41 Relationship between the analysis time and the size of the network.

Annex 1

In the Table A1, the nonautonomous formal transitor models of some electronic elements are presented. These models can be connected together and/or with autonomous models. The resulting flow-graph comprises nullators and nolators which should be removed from the network accordingly to the following rules:

1. Edges starting from nodes connected by nullator get the common starting node (one of original ones)

2. Edges arriving to nodes connected by norator get the common ending node (one of original ones)

3. Following pairs of nodes should be connected: node having only arriving edges and node having only starting edges.

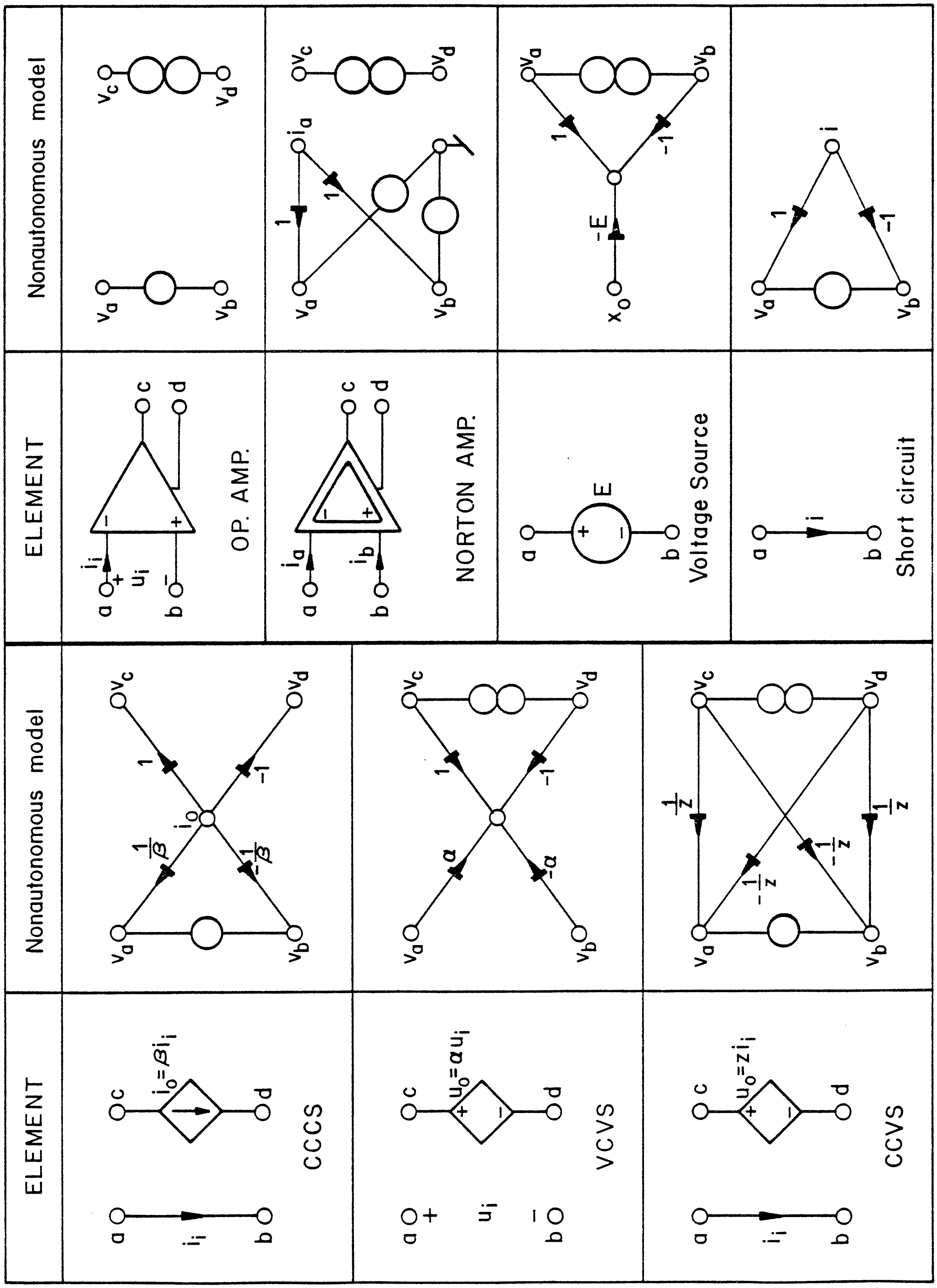


TABLE A1

NONAUTONOMOUS FORMAL TRANSITOR MODELS

SOC-306

FLOWGRAPH ANALYSIS OF LARGE ELECTRONIC NETWORKS

J.A. Starzyk and A. Konczykowska

January 1983, No. of Pages: 87

Revised:

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

Abstract: The paper presents a new complete method for signal flowgraph analysis of large electronic networks. Two efficient methods of flowgraph formation that can easily represent decomposed networks are introduced. Hierarchical decomposition approach is realized using the so-called upward analysis of decomposed network. This approach removes the limitations on topological analysis and allows to obtain fully symbolic network formulas in time which is linearly proportional to the size of the network. The approach can be used to obtain symbolic solutions of any linear system of equations.

Description:

Related Work: SOC-244, SOC-266, SOC-268, SOC-269, SOC-273, SOC-285, SOC-300.

Price: $ 12.00.