The Modified Nodal Approach to Network Analysis

Article *in* IEEE Transactions on Circuits and Systems · July 1975

DOI: 10.1109/TCS.1975.1084079 · Source: IEEE Xplore

Albert E Ruehli
Missouri University of Science and Technology
275 PUBLICATIONS 10,988 CITATIONS

The Modified Nodal Approach to Network Analysis

CHUNG-WEN HO, MEMBER, IEEE, ALBERT E. RUEHLI, MEMBER, IEEE, AND PIERCE A. BRENNAN

Abstract—**The nodal method has been widely used for formulating circuit equations in computer-aided network analysis and design programs. However, several limitations exist in this method including the inability to process voltage sources and current-dependent circuit elements in a simple and efficient manner. A modified nodal analysis (MNA) method is proposed here which retains the simplicity and other advantages of nodal analysis while removing its limitations. A simple and effective pivoting scheme is also given. Numerical examples are used to compare the MNA method with the tableau method. Favorable results are observed for the MNA method in terms of the dimension, number of nonzeros, and fill-ins for comparable circuit matrices.**

I. Introduction

The method by which circuit equations are formulated is of key importance to a computer-aided circuit analysis and design program for integrated circuits. It affects significantly the set-up time, the programming effort, the storage requirements, and the execution speed of the computer program. The method which one selects needs to be flexible, computationally efficient, and economical with storage. The nodal approach for formulating circuit equations is a classical method which not only meets these requirements but also yields a numerically well-behaved diagonal. It is therefore very popular and has been widely used in modern computer programs such as CANCER [1], ECAP [2], and BIAS-3 [3]. However, in its basic form it treats voltage sources inefficiently and is incapable of including current-dependent elements, linear or nonlinear. Therefore, several attempts have been made in the past to generalize the method. For example, in the CANCER program, each of the independent voltage sources is replaced by Norton equivalent current sources across every branch connected to the positive node of the voltage source. Calahan [4] used gyrators to convert linear and nonlinear inductors to capacitors in the time domain. Furthermore, extremely small or negative resistances [5] were introduced in some programs to accommodate current dependencies. Another disadvantage of the nodal method is that branch currents are not accurately or conveniently obtained as part of the output of the program. These complications may have contributed to the fact that other methods were implemented in the ECAP II [6] and ASTAP [7] programs.

Here, a set of self-consistent modifications to the nodal method are proposed and the resultant formulation is called the modified nodal analysis (MNA). The MNA resolves all the above-mentioned problems in the nodal method while preserving its advantages.

In Section II, some details of the MNA method are presented while an appropriate matrix ordering or pivoting scheme is discussed in Section III. Section IV gives practical examples in which the impact of the equation formulation upon the set-up time, execution speed, and storage requirements of the equations are discussed. The results are then compared to a tableau type method. Finally, Section V is devoted to discussion and conclusions.

II. Modified Nodal Approach

In formulating the circuit equations by using the MNA for a given network, we first start with the set of nodal voltages versus a common datum node as variables as is the case in the basic nodal method. Kirchhoff's current law is applied to each node other than the datum node in the circuit such that the summation of currents leaving the node is equal to zero. For the simple case of a circuit containing only linear conductances and independent current sources, the MNA generates the same set of equations as in the nodal formulation

$$Y\underline{V} = \underline{J} \tag{1}$$

where Y represents the node admittance matrix, $\underline{V}$ the common datum voltages, and $\underline{J}$ the current source vector. For the circuits which contain voltage sources and other elements whose currents are controlling variables, the MNA proceeds by introducing those branch currents as additional variables and the corresponding branch constitutive relations as additional equations. These branch currents are available as additional output variables. This is illustrated first for the example network shown in Fig. 1. Two currents I_E and I_3 are catenated to the unknown vector as is shown in Fig. 2 and branch relations are introduced for the voltage source and the nonlinear conductance. The nonlinearity in this example is expressed in terms of a Newton–Raphson iteration scheme, e.g., [10].

The MNA matrix can in general be expressed in the form

$$\begin{bmatrix} Y_R & B \\ C & D \end{bmatrix} \begin{bmatrix} \underline{V} \\ \underline{I} \end{bmatrix} = \begin{bmatrix} \underline{J} \\ \underline{F} \end{bmatrix} \tag{2}$$

where Y_R is a reduced form of the nodal matrix excluding the contributions due to voltage sources, current controlling elements, etc. B contains partial derivatives of the Kirchhoff current equations with respect to the additional current

Manuscript received October 15, 1974; revised July 19, 1974.
The authors are with the IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y.

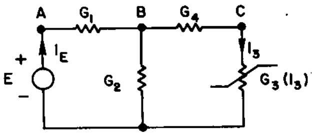


Fig. 1. Example network.

$$\begin{bmatrix} G_1 & -G_1 & 0 & -1 & 0 \\ -G_1 & G_1+G_2+G_4 & -G_4 & 0 & 0 \\ 0 & -G_4 & G_4 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & G_3 & 0 & -1+\frac{\partial G_3}{\partial I_3} V_C \end{bmatrix} \begin{bmatrix} V_A \\ V_B \\ V_C \\ I_E \\ I_3 \end{bmatrix} =$$

Fig. 2. Modified nodal equation for example network.

variables and thus contains ±1's for the elements whose branch relations are introduced. The branch constitutive relations, differentiated with respect to the unknown vector, are represented by the matrices C and D. It is noted that the zero–nonzero pattern of C is basically the same as that of B^T except for some nonreciprocal elements. The vectors $\underline{J}$ and $\underline{F}$ are excitations which include the initial values from previous time steps corresponding to capacitors and inductors which will be discussed in more detail later. It is advantageous to consider the contributions of each circuit element to the MNA matrix separately. Specifically, the "element rubber stamps" given in Table I corresponding to the circuit of a "general node" shown in Fig. 3 summarize the contributions for each type of element, where *BR* refers to the additional branch relation and RHS is the contribution to the right side of (2). The MNA matrix can be generated by using the element stamp table in a straightforward manner. For a given circuit, the matrix dimension is simply the sum of the number of nodes excluding the ground node plus the number of currents as outputs. We first arbitrarily label every node in some order then continue to label every element whose current is one of the outputs. We then collect all the elements in the network and process them one by one. For each element, depending on whether its current is an output or not, its contributions to the matrix can simply be read from the table and stamped into the matrix according to its node number or current number just labeled. The matrix in (2) is obtained when all elements are processed.

Capacitances and inductances are considered only in the time domain and their contributions, shown in Table I, are obtained by applying finite differencing methods to their branch relations. For example, the time derivative for the capacitor current $i_c(t) = C(dv_c/dt)$ is represented in terms of an implicit integration scheme [10] as

$$i_c(t_n) = \frac{C}{h}[v_c(t_n) - v_{cp}] \tag{3}$$

TABLE I
STAMPS FOR NETWORK ELEMENTS

Element Type	Element Stamps									
	Current not output					Branch current output				
G		V_i	V_1		RHS		V_i	V_1	I_G	RHS
	i	G	−G			i			1	
	1	−G	G			1			−1	
						BR	G	−G	−1	
C		V_i	V_3		RHS		V_i	V_3	I_C	RHS
	i	$\frac{C}{h}$	$-\frac{C}{h}$		$+\frac{C}{h} V_{cp}$	i			+1	
	3	$-\frac{C}{h}$	$\frac{C}{h}$		$-\frac{C}{h} V_{cp}$	3			−1	
						BR	$\frac{C}{h}$	$-\frac{C}{h}$	−1	$\frac{C}{h} V_{cp}$
L							V_i	V_5	I_L	RHS
						i			+1	
						5			−1	
						BR	1	−1	$-\frac{L}{h}$	$-\frac{L}{h} I_{1p}$
E							V_i	V_4	I_E	RHS
						i			+1	
						4			−1	
						BR	1	−1		E
J		V_i	V_2		RHS		V_i	V_2	I_J	RHS
	i				−J	i			1	
	2				+J	2			−1	
						BR			1	J

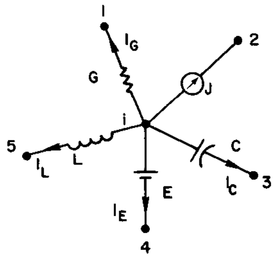


Fig. 3. General node.

where h is the time step, and v_{cp} is a function of $v_c(t)$ at previous time points. Equation (3) can thus be represented by a conductance of C/h in parallel with a current source $C(v_{cp}/h)$. The element stamps for a capacitor C in Table I are thus found by using (3). The stamp is different depending on whether the capacitor current is selected as an unknown variable or not.

General aspects for the derivation of stamps are discussed next. The branch current is always introduced as an additional variable for *a voltage source*, either independent or dependent, and *an inductor* and is thus readily available as an output variable. For current sources, resistors, conductances, and capacitors, this is only done under the following conditions:

1) if other nonlinear circuit elements depend on its current; and
2) if the branch current is requested as an output variable.

The stamps given in Table I corresponding to Fig. 3 have been derived using these considerations. Having determined the proper set of MNA equations as discussed previously, nonlinear circuit elements are first processed as linear elements, as shown in Table I. The partial derivatives of those elements versus their controlling variables contribute additional nonzeros in the MNA matrix. They are handled in a similar fashion as the example in Fig. 1 and hence are not discussed further. It is to be emphasized, however, that for most practical circuits, the number of additional variables and equations introduced, i.e., number of voltage sources, inductors, etc., is small compared to the number of nodes. The resultant set of variables and equations is thus large enough to include all useful information and yet small enough to make the formulation efficient.

III. Pivot Ordering Strategy

The MNA equations are ordered in such a way that both execution time and storage requirements are small. In principle, a general pivoting strategy similar to the one used in OPTORD [8] could be applied. However, the nodal method yields a strong diagonal which is well suited for pivoting down the diagonal from both an accuracy and efficiency point of view [1], [4]. In choosing a pivoting strategy for the modified nodal matrix we want to maintain this strong diagonal.

Consider the zero–nonzero pattern for the MNA matrix in Fig. 4 which represents a 15 transistor, 25 node *J–K* flip-flop [13] before pivot ordering. For completeness, the right-hand side of (2) is catenated to the matrix as the last column [11]. In the pivoting scheme discussed below, we may assume that the last row and column are removed. Pivoting is always carried out in the three steps discussed below, which are the processing of singletons, row interchange, and pivoting on the diagonal.

A. Processing of Singletons

We begin by choosing row and column singletons (rows and columns with only one element in them) as the initially assigned pivots. These singletons are due to either voltage sources having one terminal grounded or due to current sources. They are circled in the example matrix Fig. 4 for identification purposes.

The singletons are well suited for pivoting since their numerical values are ±1 and they do not introduce any fill-ins. Furthermore, off-diagonal singletons always appear in pairs located symmetrically with respect to the diagonal for the MNA matrix, as can be seen from Table I or Fig. 4. This is necessary for these singletons since pivoting on an off-diagonal element (and therefore, removing the corresponding row and column from the matrix) shifts the diagonal from its original position. Then pivoting on its mirror image restores it to its original position. Thus, after all singletons are designated as pivots, the new diagonal remains as part of the original strong diagonal. The procedure, therefore, is to first choose all row and column singletons with the column singletons pivoted first and the row singletons reserved to be pivoted last.

$$
\begin{array}{cccccccccccccccccccccccc|cccccc}
1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\text{\textcircled{1}}&0&0&0&0\\
1&1&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\text{\textcircled{1}}&0&0\\
0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\text{\textcircled{1}}&0\\
0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&1&1&0&0&0&0&0&1&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&1&1&0&0&0&1&1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1\\
0&1&0&0&0&0&0&0&0&0&1&1&1&1&1&0&0&1&0&1&0&1&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&1&0&0&1&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&1&0&0&1&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&\text{\textcircled{1}}&0&0&0\\
0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&1&0&1&0&0&1&0&0&0&0&0&1&1&0&1&0&1&1&\text{\textcircled{1}}&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&0&0&1&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&1&0&1&0&0&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&1&1&0&0&0&0&0&0&0&1\\
0&0&0&1&0&0&0&1&0&0&1&0&0&0&0&0&0&1&1&0&0&0&1&0&0&0&0&0&0&1\\
0&0&0&0&0&1&0&0&0&1&0&0&1&0&0&0&0&1&0&0&1&0&0&1&0&0&0&0&0&1\\
\hline
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\text{\textcircled{1}}&0&0&0&0&0&0&\boxed{0}&0&0&0&0&1\\
\text{\textcircled{1}}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\boxed{0}&0&0&0&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\text{\textcircled{1}}&0&0&0&0&0&0&0&0&0&0&\boxed{0}&0&0&1\\
0&0&\text{\textcircled{1}}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\boxed{0}&0&1\\
0&0&0&0&\text{\textcircled{1}}&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\boxed{0}&1\\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&\text{\textcircled{1}}
\end{array}
$$

Fig. 4. Modified nodal matrix before pivoting.

B. Row Interchange

At this point, all singletons have been assigned as pivots and are removed from considerations. The diagonal of the remaining matrix is preserved as part of the original one. Next, certain rows are interchanged before further pivots are selected. As shown in Table I, branch relations are always introduced for voltage sources and inductors. Voltage sources with one grounded terminal are not of interest since they have been taken care of by processing the singletons. The rest of the voltage sources and inductors do not contribute to the diagonal positions in their respective nodal equations. However, in each row of the branch relations they do contribute a +1 and a −1 in the columns corresponding to the diagonal positions of the respective nodal equations. Thus a ±1 will appear on the diagonal positions of both interchanged rows if the branch relation is interchanged with either of the nodal equations. For example, in the element stamp corresponding to L in Table I, interchanging row 5 with row BR results in a −1 in both of the new diagonal positions.

The interchange process is applied to the node equation of each node connected to one or more of the voltage sources (E's) and inductors (L's). The node equation is always interchanged with one of the branch relations of those E's and L's having the smallest number of nonzeros to minimize fill-ins. Row exchange leads to ±1 in the diagonal positions and generally results in a reduction of fill-ins since the number of nonzeros in rows corresponding to the branch relation is usually small. Further, complete numerical cancellation for pivoting along the diagonal is avoided. This may occur in the case where R, G, or C (without current specified) is connected between L's or E's only. A similar situation may occur if each of the above mentioned L's and E's is replaced by another type of element with its current specified as an output. However, this situation is not of practical interest.

C. Pivoting on Diagonal

The remaining pivot elements are now chosen from the diagonal terms of the processed matrix using a Markowitz type criteria to minimize the number of fill-ins [12]. Equations with empty diagonal terms, e.g. branch relations for voltage sources not selected to interchange with node equations are temporarily excluded from contention. However, because of the way the stamp corresponding to voltage sources is defined in Table I, a fill-in is guaranteed to take place in the diagonal position and the element can then be considered as a candidate for subsequent pivot selection. The row count r and the column count c are found for all pivot candidates and a weight of (cr) is associated with the pivots. The next pivot is the one with the smallest weight (ties are arbitrarily broken) and the matrix is reduced by elimination of the pivot row and column. The process is repeated until all pivots are found. Fig. 5 illustrates the zero–one structure of the MNA matrix after pivoting and fill-ins with the fill-in elements indicated by circles.

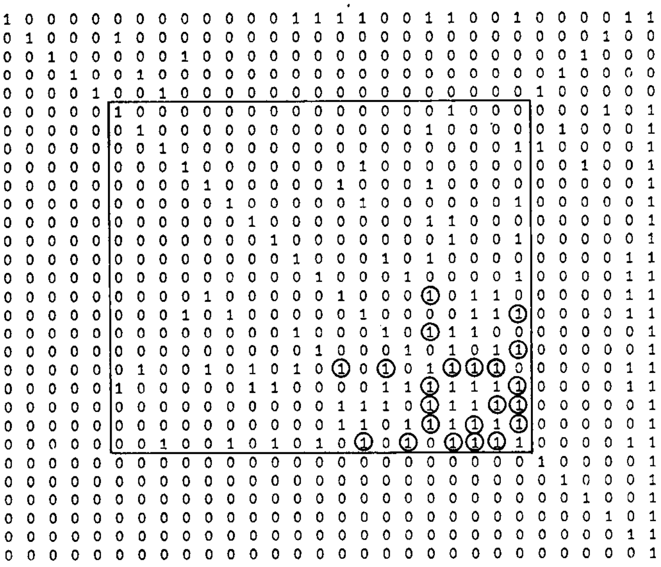
IV. Results and Comparison

Three major areas of importance contributing to the overall efficiency of a circuit analysis and design program are the set-up time, storage requirements, and execution speed. The basic network equation formulation impacts all three areas. Specifically, the matrix size and row and column counts are formulation dependent with different formulation methods resulting in different numbers of fill-ins even if the same pivoting strategy is used. The method of equation formulation impacts the set-up time which includes forming the circuit matrix, selecting the pivots, and generating the code for solving the linear equations. Since both storage requirements and execution time depend on the matrix dimension and the number of nonzeros in the matrix they are also dependent on the formulation method.

A comparison has been made between a new version of the tableau formulation (TA) [9], presently one of the more sophisticated techniques available, and the MNA method to evaluate the relative merits of the approaches. In TA, the topological relations of cut-set and fundamental circuit equations are arranged in such a way that the left half of the corresponding matrix is an identity matrix. The lower half of the matrix, which represents the branch constitutive relations, is processed by pivoting on the unity matrix. A partial pivoting scheme of the Markowitz type [12] is used for the lower half of the matrix. For the comparison, the MNA approach discussed in the previous section has been implemented in an experimental computer program. The two approaches are applied to four switching type circuits and the results are summarized in Table II. Circuit 1 is the *J–K* flip-flop [13] corresponding to Figs. 4 and 5. Circuit 2 is a differential switching circuit shown in Fig. 6 while circuit 3 represents three cascaded sections and circuit

TABLE II
Comparative Matrix Data for Four Example Circuits

	Circuit 1		Circuit 2		Circuit 3		Circuit 4	
	TA	MNA	TA	MNA	TA	MNA	TA	MNA
No. of nodes	25		46		33		43	
No. of branches	84		154		122		162	
Matrix dimension	168	29	308	50	244	34	324	44
No. of contributions to matrix	599	249	1171	512	760	439	1012	580
No. of nonzero matrix elements	599	108	1171	217	760	150	1012	198
No. of fill-ins created	295	22	719	52	370	12	501	18
Nonzero elements after fill-ins	894	130	1890	269	1130	162	1513	216



```
1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 0 0 1 0 0 0 0 1 1
0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 1
0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 1
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 ① 0 1 1 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 0 0 1 1 ① 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 ① 1 1 0 0 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 1 ① 0 0 0 0 0 1
0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 ① 0 ① 0 1 ① ① ① 0 0 0 0 0 1 1
0 0 0 0 0 1 0 0 0 0 0 1 1 0 0 0 0 1 1 ① 1 1 1 ① 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 ① 1 1 ① ① 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 ① 1 ① 1 ① 0 0 0 0 0 1
0 0 0 0 0 0 0 1 0 0 1 0 1 0 1 0 ① 0 ① 0 ① ① ① 1 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
```

○ INDICATES FILL-IN

Fig. 5. Modified nodal matrix after pivoting with fill-ins.

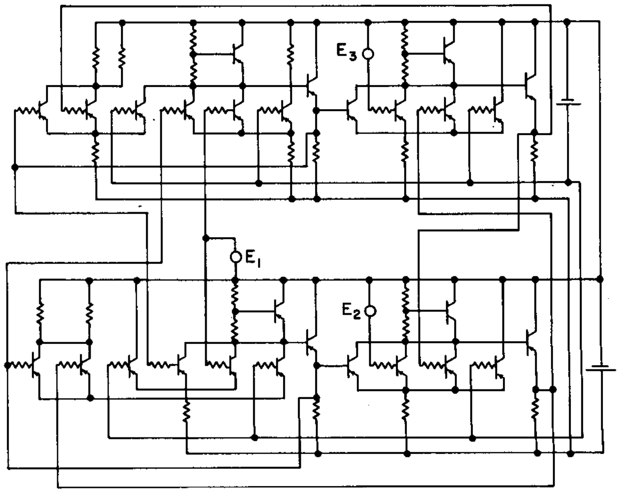


Fig. 6. A 154-branch 46-node circuit.

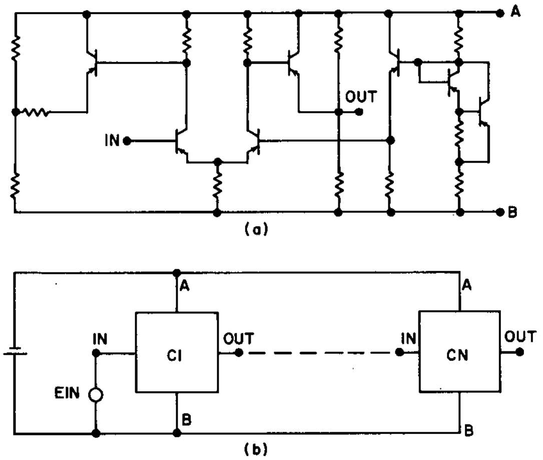


Fig. 7. (a) Basic circuit C1. (b) Cascaded basic circuits.

4 is four cascaded sections of the basic current switch circuit in Fig. 7(a). A four-element Ebers–Moll intrinsic model is used for the transistors.

As can be observed in Table II, the dimension of the MNA matrix is a factor of 5–7 smaller than the TA matrix for the four example circuits. The number of nonzeros entering the matrix for the MNA is about a factor of 2 less than the TA. However, since nonzeros are combined in the MNA matrix even before matrix processing, while every nonzero value enters into a different position in the TA matrix, the resultant number of nonzeros in the MNA matrix becomes a factor of 5 less than that of the TA matrix. This substantially reduces the processing time for pivot selection, pointers, or machine code generation for the MNA method. An even larger ratio is observed for the numbers of fill-ins for the two approaches corresponding to the four examples. Specifically, they differ by more than a factor of 10 for all cases. A contributing factor is the scheme used for preprocessing row and column singletons such that the original MNA matrix is reduced to a smaller irreducible block which results in fewer possible fill-ins. The total number of nonzeros for the two approaches after fill-ins differ by a factor of about 7.

The exact set-up time, storage requirement, and execution speed of a computer-aided circuit analysis and design program depends also on the programming implementation and other numerical techniques which are not discussed in the present paper. However, the numerical results given previously should be sufficient evidence to indicate that the MNA has a favorable impact on the performance of such a program.

V. Discussion and Conclusions

Several examples have been given in this paper for testing the MNA. For the circuits analyzed, a smaller matrix results than that obtained from the tableau type of formulation as is to be expected. More importantly, the number of nonzero elements after fill-ins is about a factor seven less in the MNA matrix. The difference between the two formulations will be reduced if a large number of branch currents are requested as outputs. However, this is not the case for most practical applications.

The following conclusion can be drawn from the results obtained from the experimental MNA program. First, the classical nodal approach can be extended such that its disadvantages are removed and its usage becomes completely general. Secondly, the set-up time spent and the amount of storage required for matrix pivoting, sparse matrix pointer, code generation, etc., are considerably reduced by the MNA compared to a tableau-type approach. Finally, for the execution speed required in solving linear equations, there are two cases. For large network whose SOLVE code [8] may require too much core for the computer at hand, the recursive Gaussian elimination type of sparse matrix method suggested by Chang [14] should be used. For this case, since the MNA method generates a much smaller matrix and less nonzeros, it is much faster than the TA method because the execution speed is directly proportional to the dimensions and nonzeros of the matrix. On the other hand, if the Crout algorithm is used for the SOLVE code generation for solving sparse matrices, the variability type concept [8] can be applied to optimize the code in terms of speed. As a result, the difference in execution speed for different formulations is not very clear. As reported by Hachtel *et al.* [8] in their example illustrated in their Table VII, a tableau matrix is first set up. Two pivoting schemes were used. One is the optimized pivoting order and the other is to constrain the pivoting order to reduce the matrix to a nodal matrix first then proceed from there. The results show that the execution speeds for the two pivoting schemes are very close. (The optimized pivoting is about 2 percent faster if we assume that for every time step there are, on the average, two Newton iterations.) However, if, as is proposed in this paper, the nodal matrix is set up directly instead of starting from a tableau and then reducing it to the nodal matrix, the execution speed will improve because some intermediate load and store instructions cannot be eliminated by the optimized complier. At present, no quantitative results have been obtained for this speed improvement.

References

[1] L. Nagel and R. Rohrer, "Computer analysis of nonlinear circuits excluding radiation (CANCER)," *IEEE J. Solid-State Circuits*, vol. SC-6, pp. 166–182, Aug. 1971.

[2] "1620 electronic circuit analysis program [ECAP] [1620-EE-02X2] user's manual," IBM Application Program File H20-0170-1, 1965.

[3] W. J. McCalla and W. G. Howard, Jr., "BIAS-3: A program for the nonlinear dc analysis of bipolar transistor circuits," *IEEE J. Solid-State Circuits*, vol. SC-6, pp. 14–19, Feb. 1971.

[4] D. A. Calahan, *Computer-Aided Network Design.* New York: McGraw-Hill, 1972.

[5] R. Engelhardt and M. Heinz, "Behandlung idealer gesteuerter Quellen bei der Netzwerkanalyse mit Digitalrechnern", NTZ, Heft 1, pp. 8–10, 1970.

[6] F. H. Branin, G. R. Hogsett, R. Lunde, and L. E. Kugel, "ECAP II—An electronic circuit analysis program," *IEEE Spectrum*, vol. 6, no. 6, June 1971.

[7] IBM advanced statistical analysis program, ASTAP, general info. manual GH20-1271-0.

[8] G. D. Hachtel, R. K. Brayton, and F. G. Gustavson, "The sparse tableau approach to network analysis and design," *IEEE Trans. Circuit Theory*, vol. CT-18, pp. 101–113, Jan. 1971.

[9] W. T. Weeks, A. Jimenez, C. W. Mahoney, H. Qassemzadeh, and

T. R. Scott, "Network analysis using a sparse tableau with tree selection to increase sparseness," *IEEE Proc. Inst. Symp. Circuit Theory*, Toronto, Ont., Canada, pp. 165–168, Apr. 1973.

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

[11] G. D. Hachtel, "Vector and matrix variability type in sparse matrix algorithms," *Sparse Matrices and Their Applications*, D. J. Rose and R. Willoughby, Eds. New York: Plenum Press, 1972, p. 55.

[12] H. M. Markowtiz, "The elimination form of the inverse and its application to linear programming," *Management Sci.*, pp. 255–269, Apr. 1957.

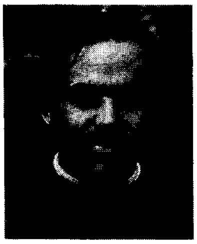
[13] W. J. McCalla and W. G. Howard, "BIAS-3": A program for the nonlinear dc analysis of bipolar transistor circuits," *IEEE Int. Solid-State Circuit Conf. Dig.*, pp. 82–83 and p. 187, 1970.

[14] A. Chang, "Applications of sparse matrix methods in electric power system analysis," *Proc. Sparse Matrix Symp.*, IBM Rep. RA-1, pp. 113–121, Mar. 1969.

Chung-Wen Ho (S'63–M'66) was born in An Huei, China, on September 8, 1940. He received the B.S. degree in electrical engineering from National Taiwan University, Taipai, Taiwan, China, in 1961 and the M.S. and Ph.D. degrees in electrical engineering from the University of California, Berkeley, in 1963 and 1966, respectively.

He joined the staff of the Electronics Research Laboratory, University of California, Berkeley, as a Research Assistant in 1963. In 1966, he became an acting Assistant Professor in the Department of Electrical Engineering, University of California, Berkeley. Since 1967, he has been a Member of the Research Staff, Thomas J. Watson Research Laboratory, IBM Corporation, Yorktown Heights, N.Y. His current interests include computer aided circuits analysis and design, and the design of microprocessors.

Dr. Ho received the Best Paper Award in IEEE, Syracuse, N.Y., Chapter, in 1967.



Albert E. Ruehli (M'65) was born in Zürich, Switzerland, on June 22, 1937. He received the Telecommunication Engineering degree in 1963 from Abend Technicum, Zürich, and the Ph.D. degree in electrical engineering in 1972 from the University of Vermont, Burlington.

From 1958 to 1963, he worked at the IBM Research Laboratory, Zürich, on thin magnetic film memories. In 1963, he transferred to the IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., where he worked in the area of semiconductor circuits and devices. In 1966, he moved to IBM, Burlington, where he was associated with the mathematical and engineering analysis group at the Components Division Laboratory. Since 1971, he has been a Research Staff Member at the Thomas J. Watson Research Center, where he is in the Circuit and Computer Design Automation Group developing new techniques for analysis of the electrical parameters of computer hardware technologies.

Pierce A. Brennan was born in New York City, N.Y. He received the A.B. and M.S. degrees in physics from Fordham University, New York, N.Y., in 1948 and 1950, respectively.

From 1948 to 1949, he was engaged in radar receiver design and, from 1950 to 1956, in microwave tube research at the Evans Signal Laboratory, Belmar, N.J. In 1956, he joined the Electron Tube Laboratory at Stanford University, Stanford, Calif., where he continued work in the microwave tube field. Since 1960, he has been at the IBM Thomas J. Watson Research Center, Yorktown Heights, N.Y., where he has been concerned with packaging problems associated with high-speed circuit technology and, most recently, the area of circuit-design automation.

Mr. Brennan is a member of the American Physical Society, the American Association of Physics Teachers, and Sigma Xi.

Low-Transient Intermediate-Band Crystal Filters

HERMAN J. BLINCHIKOFF, SENIOR MEMBER, IEEE

Abstract—**A general technique for obtaining intermediate-band crystal filters from prototype low-pass (LP) networks which are neither symmetric nor antimetric is presented. This immediately enables us to now realize the class of low-transient responses. The bandpass (BP) filter appears as a cascade of symmetric lattice sections, obtained by partitioning the LP prototype filter, inserting constant reactances where necessary, and then applying the LP to BP frequency transformation. The cascade is composed of only two fundamental sections. Finally, the method introduced is illustrated with an example.**

I. Introduction

Bandpass (BP) crystal filters are usually divided into the following three bandwidth classes according to their design.

1) Narrow band: This class comprises relative bandwidths less than about 0.2 percent; no loading inductors are included [1]–[3].

Manuscript received January 7, 1974; revised October 9, 1974.

The author is with the Systems Development Division, Westinghouse Electric Corporation, Baltimore, Md.