Two Scattering Matrix Programs for Active Circuit Analysis

PHICHANI BODHARAMIK, LES BESSER, MEMBER, IEEE, AND ROBERT W. NEWCOMB, MEMBER, IEEE

Abstract—What's a diamond to an ape, do you think?
It's nothing to be eaten by a monkey.

—*Thai poem*

The philosophy is advanced of having two programs to cover a given area, one for quick routine calculations and one to handle general cases. Thus, two scattering matrix programs are described for the area of computer-aided analysis of active circuits. Both programs rest upon an iterative use of cascade loading, a connection previously proven to incorporate any arbitrary connection. One program, GENERAL, covers the multiport case and through an incorporation of the generality of the scattering matrix handles any physical network. GENERAL proceeds by finding the scattering matrix of interconnections through the use of the indefinite admittance and by iteratively cascade loading such interconnections with components. The other (time-shared) program, SPEEDY, is a fast 2-port version. SPEEDY's time advantage comes about by programming the specific results for the most typical design connections (to which the program is limited). These are the series–parallel and cascade connections for 2-ports interpreted in terms of scattering parameters. SPEEDY contains two unique subroutines: one for all possible 2-port connections and one for converting common emitter scattering parameters to those for common base or collector.

Scant knowing much knows	Vain heart
Frog at birth you start	Small pond
Short of sight, apart	Mid-sea
Be of that pond fond	Wide very deep [1, p. 7].

I. Introduction

FOR SOME time the scattering matrix has been recognized as a powerful tool in circuit theory [2], [3]. The reasons for this are manifold; for example, general syntheses and a complete theory of equivalence for finite networks can be given in terms of the scattering matrix [4]. Indeed, passive, as well as active, networks generally possess a scattering matrix [5, p. 90], a property not held by other familiar descriptions. These advantages, of course, carry over into the area of computer-aided analysis, where there appears to be an interest developing in the computational uses of the scattering matrix [6], [7].

It is interesting to note that some previous microwave circuit analysis routines accepted scattering matrix inputs [8]–[10], but with small exceptions [11], [12], calculations have been made using other descriptions. Since measurements can now be made directly to yield scattering parameters, using, for example, the HP 8410 network analyzer system [13], it would seem inefficient to be constantly converting to other descriptions. This is especially true in the areas of active and transistorized circuits where manufacturers are beginning to furnish data in terms of polar plots of scattering parameters well into the gigahertz region [14], [15]. Since equivalent circuits based upon admittance matrices break down at these frequencies, and since scattering matrix design methods have been introduced [16], [17], it seems imperative to have solid routines available for analysis in terms of scattering parameters. Such routines would be particularly valuable for checking designs and interconnections of integrated circuits for UHF use, since, as mentioned above, the scattering matrix has presented the most valid description at these frequencies.

Here we present the theory and outline the operational details behind two such routines. One program, SPEEDY, is set up for very rapid analysis of 2-ports; it is programmed for the conversational time-shared mode on the SDS Sigma 7 computer and is limited to those 2-ports which can be constructed as a sequence of series–parallel–cascade connections of 2-ports. The other program, GENERAL, is set up for n external ports, $n \leq 30$, on the UNIVAC 1108 computer using batch processing; it can handle arbitrary internal connections through a sequence of connections of m-ports, $m \leq 30$. In principle, GENERAL can be extended to cover any network, its capability being limited only by the storage capacity available.

Both programs are based upon the general cascade-load connection which previously [18] has been shown to include all other connections as special cases. The concept of cascade loading allows the development, in Section III, of a general formula for series–parallel connections. This is applied in SPEEDY, while the augmented admittance calculated through indefinite matrices is applied in GENERAL. Flow charts for both GENERAL and SPEEDY are given at the ends of Sections V and VI, respectively. Cascade loading is reviewed next in Section II, as well as some basic scattering matrix formulas.

II. Cascade Loading

Here we develop the main formulas on which the programs are based. In particular we review the ideas behind, and the results of, cascade loading.

Because physical measurements of scattering matrices are most often taken with regard to nonunit terminations, we begin with the $n \times n$ scattering matrix S_R of an n-port N with

Manuscript received February 3, 1971; revised June 1, 1971. This work was supported in part by the National Science Foundation, under Grants GK1956 (Stanford) and GK24036 (Maryland), and in part by the Air Force Office of Scientific Research, under Contract AFOSR F44620-67-C-0001 (Stanford) and under Grant 70-1910 (Maryland).

P. Bodharamik is with the Department of Electrical Engineering, University of Maryland, College Park, Md. 20742.

L. Besser is with Fairchild Microwave and Optoelectronics Division, Palo Alto, Calif. 93404.

R. W. Newcomb was with Stanford University, Stanford, Calif. 94305. He is now with the Department of Electrical Engineering, University of Maryland, College Park, Md. 20742.

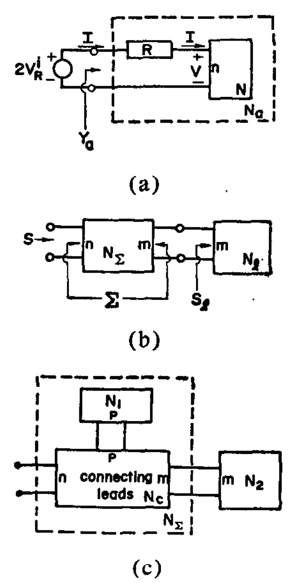


Fig. 1. Scattering matrix definitions. (a) Augmented network: $S_R = 1_n - 2RY_a$. (b) Cascade-load connection. (c) Iterative connection of N_1 and N_2. First step: $N_\Sigma = N_c$, $N_l = N_1$. Second step: $N_\Sigma = N_c + N_1$, $N_l = N_2$.

reference to the positive definite (real, constant, $n \times n$) matrix R. This is defined by

$$V_R{}^r = S_R V_R{}^i$$

with

$$2V_R{}^i = V + RI \qquad 2V_R{}^r = V - RI \tag{1}$$

where the incident $V_R{}^i$, and reflected $V_R{}^r$, n-vector voltages are as defined in terms of true port voltages V and currents I. If the given n-port N has resistors of impedance R connected in series, then, as shown in Fig. 1(a), a new (augmented) n-port N_a is formed whose admittance matrix Y_a can be used to calculate S_R from

$$S_R = 1_n - 2RY_a. \tag{2a}$$

In (2a), 1_n denotes the $n \times n$ identity matrix. In theory, and for some calculations, it is most convenient to normalize R to 1_n in which case we simplify the notation by writing $S = S_{1_n}$. For a given network N we can convert between S and S_R through the use of the equations [5, p. 76]

$$S = [(1_n - R^{-1}) + (1_n + R^{-1})S_R] \cdot [(1_n + R^{-1}) + (1_n - R^{-1})S_R]^{-1} \tag{2b}$$

$$S_R = [(1_n - R) + (1_n + R)S] \cdot [(1_n + R) + (1_n - R)S]^{-1}. \tag{2c}$$

Note that these are the same equation if we replace S_R by S, S by S_R, and R^{-1} by R; thus both are programmed identically. Note too that both S and S_R describe the given n-port N, though different augmented n-ports N_a are considered.

Now let a loaded m-port N_l be connected at the final m ports of a coupling $(n+m)$-port network N_Σ as shown in Fig. 1(b). Then, since incident voltages for N_l are reflected voltages for the final m ports of N_Σ, and vice-versa, we find [18, p. 971], [19, p. 151]

$$S = \Sigma_{11} + \Sigma_{12} S_l (1_m - \Sigma_{22} S_l)^{-1} \Sigma_{21}. \tag{3}$$

Here $\Sigma = [\Sigma_{ij}]$ is partitioned into submatrices Σ_{ij} according to the port decomposition of N_Σ shown in Fig. 1(b). Further, this equation holds on insertion of subscript R as long as the reference R matrix for N_l is the same as that for the final m ports of N_Σ (and that for N_Σ is a direct sum of R matrices for its input and output ports).

It can occur that the inverse indicated in (3) does not exist, but this is an extremely rare occurrence, generally indicating output ports completely isolated from the input [18, p. 972]. Hence, singular $1_m - \Sigma_{22} S_l$ will be ignored in the programming.

As shown in [18, p. 971], the cascade-load connection of Fig. 1(b) includes all possible connections as special cases. Thus, (3) is a universal equation which once programmed can be called upon iteratively. For example, consider the situation of Fig. 1(c) where a p-port N_1 and an m-port N_2 are interconnected through an $(n+m+p)$-port N_c of connecting leads. Using (3), N_1 can be considered as a load on N_c; then N_2 can be considered as a load on the resultant. Both N_1 and N_2 can be first decomposed in this same manner if desirable; then the situation of Fig. 1(c) is seen to be universal, leading to an iterative use of (3).

Basically the programs to be described represent an implementation of these ideas with attention also to calculation of scattering matrices for connecting leads and special connections.

One of these connections, of direct use later, is the cascade connection. If S_1 and S_2 are the scattering matrices of a $(k+l)$-port and an $(l+t)$-port in cascade, then interpretation of the connection as a cascade load yields, through (3),

$$S = \begin{bmatrix} S_{11_1} + S_{12_1} S_{11_2} (1_l - S_{22_1} S_{11_2})^{-1} S_{21} & S_{12_1} (1_l - S_{11_2} S_{22_1})^{-1} S_{12_2} \\ S_{12_1} (1_l - S_{11_1} S_{22_2})^{-1} S_{21_2} & S_{22_2} + S_{21_2} S_{22_2} (1_l - S_{11_2} S_{22_1})^{-1} S_{12_2} \end{bmatrix}. \tag{4}$$

We note too that when 2-ports are cascaded, only scalars are inverted in this equation, since $l = 1 = k = t$.

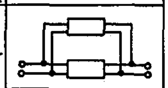
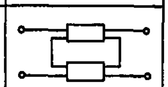
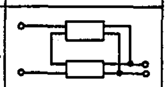
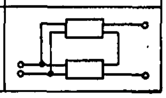
III. Series–Parallel Connections

It frequently occurs that series–parallel connections are met, as when active 2-port are designed. Since these are basic to SPEEDY, we discuss these connections in terms of scattering matrices.

We begin with the parallel connection of two n-ports, N_1 and N_2. This can be considered in cascade-load form if in Fig. 1(b) N_Σ consists of leads directly connecting input and output with N_1 directly across them [5, p. 64]. If Y_1 is the

TABLE I

E_K FOR 2-PORT K

K	E_K	CONNECTION PORT 1	CONNECTION PORT 2	CIRCUIT
Y	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$	PARALLEL	PARALLEL	
Z	$\begin{bmatrix}-1 & 0\\ 0 & -1\end{bmatrix}$	SERIES	SERIES	
H	$\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$	SERIES	PARALLEL	
G	$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$	PARALLEL	SERIES	

admittance matrix for N_1 we directly find, using variables as in Fig. 1(b),

$$\begin{bmatrix}1_n & -1_n\\ Y_1 & 0_n\end{bmatrix}\begin{bmatrix}V_1\\ V_2\end{bmatrix}=\begin{bmatrix}0_n & 0_n\\ 1_n & 1_n\end{bmatrix}\begin{bmatrix}I_1\\ I_2\end{bmatrix}. \tag{5a}$$

This is in the general description form $AV=BI$ from which $\Sigma=(B+A)^{-1}(B-A)$ is found as [5, p. 51]

$$\Sigma=\begin{bmatrix}(31_n+S_1)^{-1}(S_1-1_n) & 2(31_n+S_1)^{-1}(S_1+1_n)\\ 2(31_n+S_1)^{-1}(S_1+1_n) & (31_n+S_1)^{-1}(S_1-1_n)\end{bmatrix} \tag{5b}$$

where the following has been used to calculate S_1 from Y_1 [5, p. 52]:

$$S=(1_n+Y)^{-1}(1_n-Y). \tag{6}$$

The cascade-load equation (3) then yields for the parallel connection

$$S=\mathcal{A}^{-1}\{\mathcal{B}+4\mathcal{C}S_2[\mathcal{A}-\mathcal{B}S_2]^{-1}\mathcal{C}\} \tag{7a}$$

where the functions of S_1 are written for convenience as

$$\mathcal{A}=31_n+S_1 \quad \mathcal{B}=S_1-1_n \quad \mathcal{C}=2(1_n+S_1). \tag{7b}$$

Next let K stand for any hybrid matrix, for example, $K=Y, Z, G,$ or H in the 2-port case. Then define the matrix function $f[\cdot]$ by

$$f[K]=(1_n+K)^{-1}(1_n-K) \tag{8}$$

which is a self-inverse function;

$$f^{-1}[f[K]]=K=(1_n-f[K])(1_n+f[K])^{-1}=f[f[K]].$$

It is now true that

$$S=E_Kf[K] \tag{9}$$

where E_K is a diagonal matrix with $+1$ or -1 as diagonal entries. For example, in the $n=2$ 2-port case, E_K is chosen from Table I. Equation (9) and Table I can be verified by noting that one K matrix can be turned into another of identical numerical value by introducing unit gyrators in cascade with the ports. For example, a 2×2 Y is changed into H, with $Y=H$ numerically, if a unit gyrator is cascaded with the first port. Equation (6) gives $S_l=(1_2+H)^{-1}(1_2-H)$, while (3) yields $S=E_H f[H]$ where Σ has $\Sigma_{11}=\Sigma_{22}=0_2$, $\Sigma_{21}=1_2$, and $\Sigma_{12}=(-1)\dot{+}1$ ($\dot{+}$ means direct sum).

Next consider two n-ports N_1 and N_2 connected such that their K matrices add, that is, such that $K=K_1+K_2$; and let $S_{K_1+K_2}$ denote the scattering matrix for this connection. We have for appropriate matrix functions $g(\cdot,\cdot)$ and $h(\cdot,\cdot)$

$$\begin{aligned} S_{K1+K_2} &= E_Kf[K]=E_Kf[K_1+K_2]=E_Kg(K_1,K_2) && (10a)\\ &= E_Kg(f^{-1}[E_KS_1],f^{-1}[E_KS_2])\\ &= E_Kh(E_KS_1,E_KS_2). && (10b)\end{aligned}$$

Equations (10a) and (10b) show that the scattering matrix for connections under which K matrices add is always found from the two subnetwork scattering matrices S_1 and S_2 by using the same function $h(\cdot,\cdot)$ independent of K; the nature of the particular K considered enters only through the multiplier E_K which occurs on all scattering matrices in the formula (note also that $E_K^{-1}=E_K$). It thus remains to evaluate $h(\cdot,\cdot)$; observe that the intermediate function $g(\cdot,\cdot)$ is given by $g(K_1,K_2)=f[K_1+K_2]$.

The easiest situation in which to evaluate $h(\cdot,\cdot)$ is when $K=Y$ since E_Y is the identity matrix. But this case was handled in (7); hence

$$\begin{aligned}h(S_1,S_2)=(31_n+S_1)^{-1}\{&(S_1-1_n)+4(1_n+S_1)S_2[(31_n+S_1)\\ &-(S_1-1_n)S_2]^{-1}(1_n+S_1)\}. \end{aligned}\tag{10c}$$

In conclusion, any (series–parallel) connection for which hybrid (K) matrices add can have its scattering matrix evaluated as

$$S(S_1,S_2)=E_Kh(E_KS_1,E_KS_2) \tag{10d}$$

with $h(\cdot,\cdot)$ given by (10c); for 2-ports the E_K are all given in Table I.

IV. CALCULATION OF SPECIFIC SUBNETWORK S

Next we consider the question of actually calculating the scattering matrix of some important subnetworks used in the connections.

As seen by (9), if a hybrid matrix is originally specified, then S can be calculated; this is especially convenient for 1- or 2-port subnetworks since only simple inverses are required. Otherwise a decomposition, as in Fig. 1(c), into 1- or 2-port subnetworks may be recommended. If, however, one desires the scattering matrix for connecting leads, as for N_c of Fig. 1(c), the calculation is most easily made by finding the augmented admittance matrix Y_a and applying (2a). Indeed, Y_a will always exist for connecting leads N_c, since N_c is a linear passive time-invariant solvable multiport [18, p. 971], and Y_a is most readily found by setting up the node-by-node indefinite admittance matrix [20, p. 78] for the augmented network and then eliminating internal nodes iteratively one at a time. Internal nodes are eliminated by allowing no external current to enter them. This is equivalent to cascade loading with open circuits, which can be expressed by [5, p. 60]

$$Y=Y_{11}-Y_{12}(Y_{22}+Y_l)^{-1}Y_{21}, \qquad Y_l=0_m \tag{11}$$

where the coupling admittance is partitioned in accordance with Fig. 1(b). Alternatively, (6) could be applied to the node-by-node admittance and then the cascade-load formula of (3) used with $S_l = 1_m$; however, this often requires the inversion of a large matrix and is hence avoided.

In the cases where $R = r1_n$, with r a scalar, the R cancels out of (2a) when calculating S_R for connecting leads; that is, S_R is independent of this type of R for connecting leads. Consequently, we most conveniently set $R = 1_n$ in the programming when calculating the scattering matrix of connecting leads from Y_a.

Important 2-port subnetworks frequently arising (e.g., in ladder networks) are the series and shunt (1-port) arms in cascade. We find, on using (9), that for the series arm case

$$s_{11} = \frac{1}{1+2y} = \frac{1+s}{3-s} = s_{22} \qquad s_{12} = s_{21} = 1 - s_{11}$$

while for the shunt arm case

$$s_{11} = \frac{-1}{1+2\hat{z}} = \frac{\hat{s}-1}{3+\hat{s}} = s_{22} \qquad s_{12} = s_{21} = 1 + s_{11}.$$

Here y and $\hat{z}$ are the admittance and impedance of the 1-ports, with s and $\hat{s}$ their scattering parameters. In either case, the entire scattering matrix is completely determined, and in a manner which is simply programmable when any one entry, say s_{11}, is known. Too, s_{11} in terms of s or $\hat{s}$ can be arrived at using (10d) with $n = 1$ and $S_2 = 0$.

Transistors will frequently occur in active structures, but in several basic configurations (e.g., grounded-emitter or grounded-collector). Although the transistor is a 3-terminal device, its characteristics are most easily measured on a 2-port basis. For example, the 2×2 scattering matrix for a transistor treated as a 2-port with the emitter grounded is measured by inserting the augmenting resistors in the base and collector leads and treating the emitter common to the two ports. To convert to a 3×3 scattering matrix, for the transistor as a 3-terminal device, an augmenting resistor can be inserted in the emitter through use of a series 3-port connection via (10d). With equal augmenting resistors, $R = r1_3$, the 2×2 scattering matrix resulting from (10d) can be extended to a 3×3 matrix, S_i, by inserting a third row and column for which all resultant rows and columns sum to 1 [17, p. 7-2]. This is seen from $S_i = 1_3 - 2rY_a$, for which Y_a is indefinite (i.e., the rows and columns of Y_a sum to zero). The 3-terminal transistor is then described by S_i, which can next be converted to any 2-port configuration by appropriate cascade loading. For example, grounding the collector corresponds to shorting the collector augmentation resistor with a cascade load that is a short. The same procedure can of course be used for any 3-terminal device which is conveniently used as a 2-port in various configurations, and the ideas naturally extend to higher dimensions.

V. Specifics of GENERAL

Because the cascade-load connection incorporates every other connection as a special case, a program to implement cascade loading will have few limitations. Here we describe

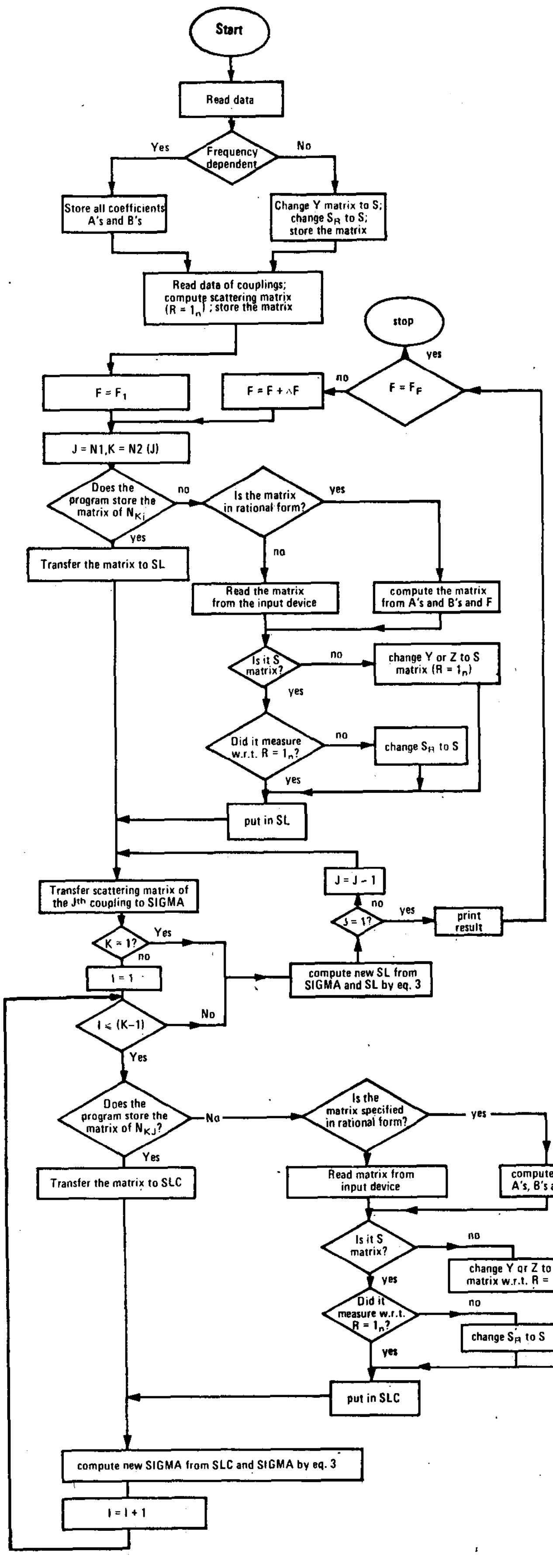


Fig. 2. Flow chart for GENERAL.

such a program, GENERAL, which iteratively applies the cascade-loading formula (3) to any decomposition of the circuit under consideration that the user may wish to make. GENERAL is then a frequency-domain analysis program designed to handle any circuit for which the scattering matrix serves as an adequate description. The program length is approximately 45 000 words; a flow chart is shown in Fig. 2.

A. General Description

The program has been written in Fortran V for the UNIVAC 1108 to compute scattering matrices of n-port networks ($n \leq 30$). As a first step in using the program, the network should be decomposed into subnetworks which form a sequence of cascade-load connections. Each subnetwork consists of one coupling network (made of connecting leads), at most four 1- or 2-port subloads (consisting usually of circuit elements), and one m-port load ($m \leq 10$); the load itself can be considered most often to be subdivided as an iteration of cascade loads. Fig. 3(b) shows an example of a network which is decomposed into three subnetworks in this manner. The first subnetwork N_{Σ_1} has one connecting lead subnetwork N_{c_1} and one subload N_{11}; everything to the right acts as a load. The second subnetwork N_{Σ_2} is similar, while the last subnetwork N_{Σ_3} has no subload in this case, but it has the load N_L (the program allows up to 10 ports in N_L). The program will compute the scattering matrix of the connecting lead coupling network through the augmented admittance matrix, as described in Sections II and IV. Then the cascade-loading connection is used to compute the overall scattering matrix of the given network through (3). Starting from the last subnetwork, the program will combine the scattering matrix of N_{Σ_3} with that of N_L. The program considers this combination as the load for the second subnetwork; the same algorithm is repeated until all subnetworks are combined to result in the final 2-port network. It should be noted that the designer has some freedom in the decomposition of the network he chooses. For example, in Fig. 3(b) one could place either or both of the transistors in the load N_L. However, experience has shown that actual computation time is relatively independent of the decomposition used, and hence there seems little preference in this regard.

B. The Input–Output

The input to the program is user-oriented. The data are fed in formatless form using the NAMELIST feature of Fortran V. The program can handle the following types of input data.

1) Rational subload Y, Z, or S matrices whose entries are of the form $(Ap^2+Bp+C)/(Dp^2+Ep+F)$. The program will store all coefficients which are assumed to be complex, at the beginning of execution. Subload matrices are assumed 1×1 or 2×2, while load matrices can have from one to ten rows and columns.

2) Measured frequency-dependent S matrices. These can be of any order and are stored in a separate file because of storage limitations. The machine calls upon this file each time this data is needed.

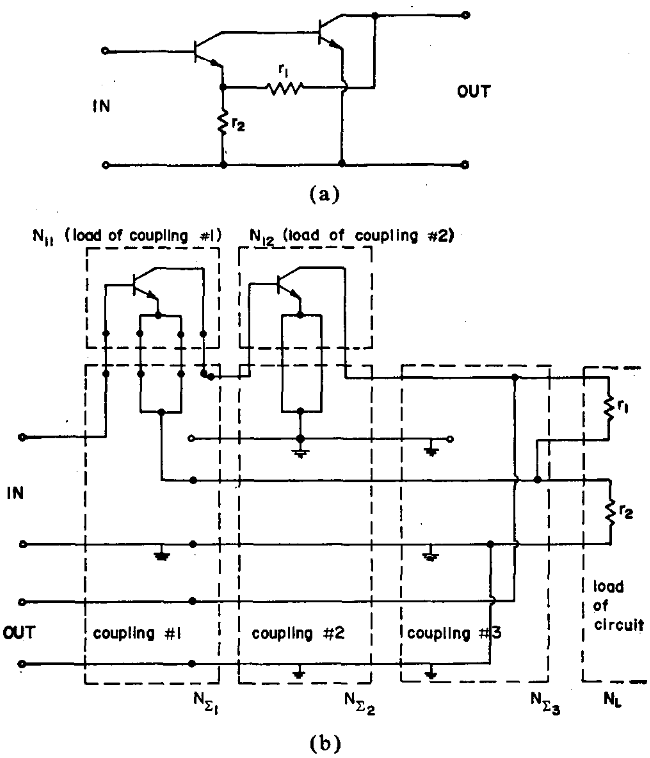


Fig. 3. (a) Two-transistor amplifier circuit diagram. (b) Decomposition for GENERAL.

3) Input data for coupling networks. These are specified in terms of number of nodes (up to 60), number of inner nodes, number of grounded nodes, and value of reference resistance [R of (1)].

C. Subroutines

The program consists of three important subroutines.

1) SYIND: This is used to compute the scattering matrix of connecting leads by setting up the node-by-node indefinite admittance matrix for the augmented network. Then internal nodes are eliminated iteratively one at a time by using (11). Because all subnetworks are specified by ports in the program, the node-by-node admittance matrix is changed to a port matrix before using (2a) to compute the scattering matrix of the connecting leads. This conversion is done by associating nodes in pairs and using voltage differences across the nodes as port variables while requiring equal currents for the pair of port terminals.

2) CPSL: This changes a Y matrix to an S matrix with respect to the reference $R = 1_n$ by using (6).

3) CSLD: This is used to combine two scattering matrices and is based on the cascade load equation (3).

VI. Specifics of SPEEDY

Although GENERAL can handle almost any circuit desired, it often occurs that such full generality is not needed. This is especially the case for most 2-port active circuits which,

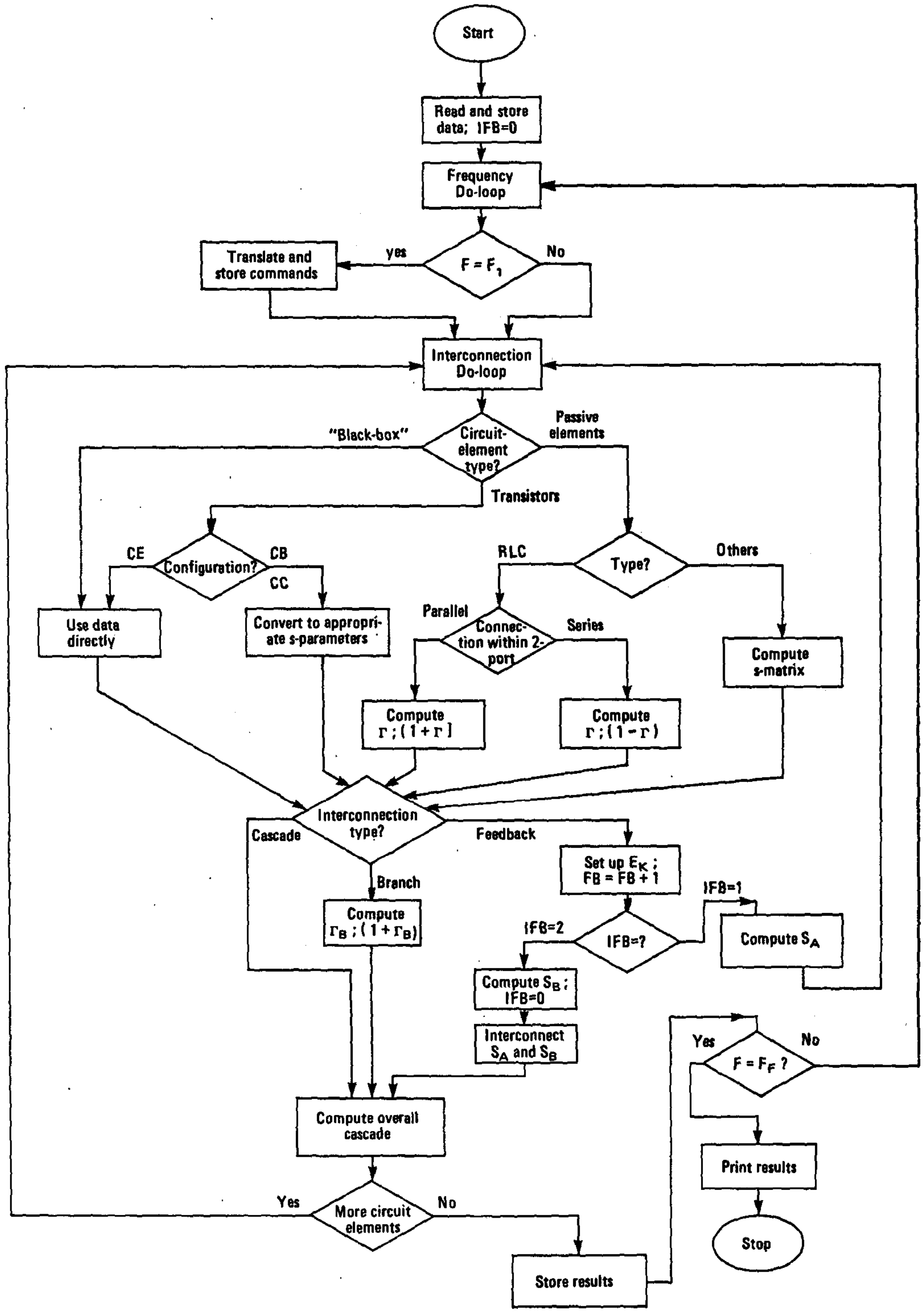


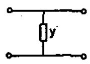
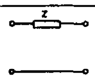
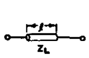
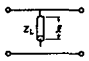
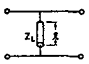
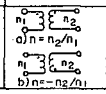
Fig. 4. Flow chart of the analysis routine SPEEDY.

in the usual situations, are designed through series–parallel or cascade connections of sub-2-ports. For analysis of this latter class of 2-ports, SPEEDY was developed as a fast economical interactive routine.

Basically SPEEDY works with (4) and (10d), thus eliminating parameter conversions. As a consequence, the circuit to be analyzed needs to be broken into combinations of 2-port connections of the type required (cascade or series–parallel). Actually the program introduces for convenience another connection command called BRANCH. This latter is used for the calculation of scattering matrices of complicated arms consisting of many elements which are inserted in

TABLE II

SCATTERING MATRICES OF CIRCUIT ELEMENTS IN SPEEDY

CIRCUIT ELEMENT		SCATTERING PARAMETER	NOTES
LUMPED SHUNT RLC		$S_{11}=S_{22}=\frac{-1}{1+2(y_0/y)}$ $S_{21}=S_{12}=1+S_{11}$	
LUMPED SERIES RLC		$S_{11}=S_{22}=\frac{1}{1+2(z_0/z)}$ $S_{21}=S_{12}=1-S_{11}$	
LOSSLESS TRANSMISSION LINE		$S_{11}=S_{22}=\Gamma_L(e^{-j2\phi}-1)/d$ $S_{21}=S_{12}=e^{-j\phi}(1-\Gamma_L^2)/d$	$\Gamma_L=\frac{Z_L-Z_0}{Z_L+Z_0}$ $\phi=\frac{\omega\cdot\ell\cdot\sqrt{E_r}}{c}$ $d=1-\Gamma_L^2e^{-j2\phi}$ E_r=rel.diele.const.
OPEN STUB		$S_{11}=S_{22}=\frac{-1}{1+2(y_0/y)}$ $S_{21}=S_{12}=1+S_{11}$	$y=j\tan(\phi)/z_L$
SHORTED STUB		$S_{11}=S_{22}=\frac{-1}{1+2(y_0/y)}$ $S_{21}=S_{12}=1+S_{11}$	$y=-j\,\mathrm{cotan}(\phi)/z_L$
IDEAL TRANSFORMER		$S_{11}=-S_{22}=(1-n^2)/d$ $S_{21}=S_{12}=2n/d$	$d=1+n^2$

series or parallel with the main signal path. Thus, calling upon BRANCH allows the previous calculations to be held while the values for the next interconnecting arm are being calculated. The program is capable of handling circuits with multiple feedback loops, where the feedback may be applied through more than one stage (overall feedback). A simplified flow graph of the analysis program is shown in Fig. 4, while Table II gives some of the scattering parameters used in SPEEDY. The major sections of the program will be discussed next.

A. Input–Output

1) *Input data file* is set up and stored by the user and is completely independent of the main program that is stored in binary form. When the main program SPEEDY, is called by the user, it asks for the name of user's data file. After receiving the information, the program proceeds with the execution. The data file is read and stored in one step at the beginning of execution. This is done to eliminate continuous jumps between the main program and the data file. The following three types of data are read and stored in one- and two-dimensional arrays.

a) *Circuit element* descriptions indicate in abbreviated from the kind of components and their positioning within the 2-port of which they are assumed a part.

b) *Interconnection commands* indicate how the particular 2-port is connected into the circuit. Three groups of such commands are recognized in abbreviated form: cascade (CASC), branch (BRAN) and series–parallel combinations (SSFB, PSFB, etc.) (K-matrix addition).

c) *Component values* or parameters include resistances, capacitances, inductances, transformer turns ratios, and transmission line descriptions. For 3-terminal devices the program accepts directly the measured 2-port S or Y parameters (generally in the common-emitter configuration for transistors).

2) *Output printout* includes the S parameters of the overall circuit and stability information if active devices are used. Optional data include return loss, VSWR, interstage mismatch losses and individual stage gains, simultaneous conjugate matches, etc.

B. Interpretation of Circuit Elements and Interconnection Commands:

For the user's convenience, these commands are given in abbreviated form and are accepted by the computer as alpha variables. During analysis for the first frequency, the alpha commands are translated to numerical codes to direct execution to the appropriate functions or subroutines by a computed GO TO statement. These codes are stored to direct execution for the following frequencies or repeated iterations.

C. Subroutines and Statement Functions

Since all the 2-terminal (1-port) elements and branch connections are described by their reflection coefficients, function statements are sufficient for computing the scattering matrices. Therefore, only a few subroutines are used in the program, mainly for computing the S parameters of the 3- and 4-terminal components (transistors, transmission lines, etc.), handling interconnections, and for optimization.

The feedback interconnection subroutine is based on (10c) and (10d). For programming convenience, the two nonzero terms of E_K are expressed as two separate variables, p and q; $E_K=\mathrm{diag}\,[p,q]$. The values of p and q for the various types of interconnections are given in Table I. When the matrix expression (10c) is replaced by its four calculated entries, some of the terms conveniently cancel and factor out for easy programming.

It has been shown earlier (Section IV) how 3-terminal scattering parameters can be computed from 2-port data. This is programmed and is especially useful when only transistor common-emitter data are available and the transistor is to be used in some other configuration.

D. Program Limitations

By its nature SPEEDY is limited to the cascade, branch, and combinations of series and parallel connections discussed earlier. However, as shown by practical experience, it will handle the majority of actual high-frequency active circuits, particularly small-signal amplifier circuits. The scope of the program could be somewhat extended by adding subroutines for special cases at a loss of convenience and speed of execution to the user.

Special care must be exercised when using parallel and series feedback combinations to see that Brune's tests [21, p. 191] and conditions for additions of K matrices are fulfilled.

E. General Information

The program was written specifically for remote time-shared operation in Fortran IV language. The length of the analysis routine is approximately 18 000 characters. SPEEDY

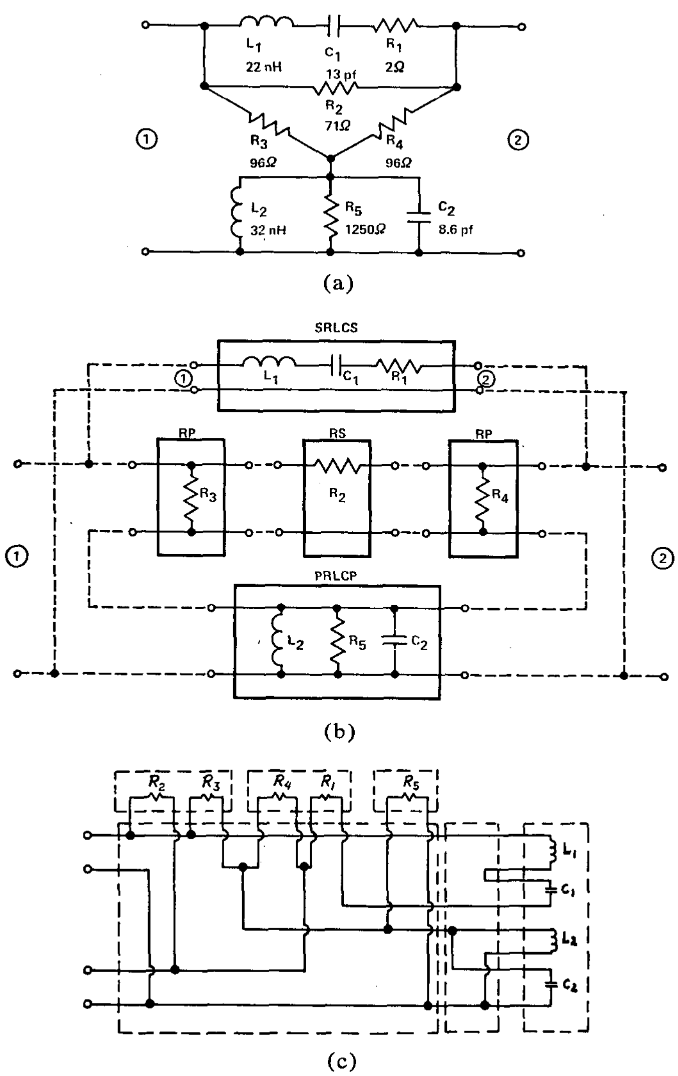


Fig. 5. (a) Circuit diagram of the modified bridged-T equalizer. (b) Decomposition for SPEEDY. (c) Decomposition for GENERAL.

has been used successfully in designing various active and passive components (including several hybrid microcircuits) of a bidirectional CATV system at Fairchild Microwave and Optoelectronics Division. One advantage is that users are not required to know any programming.

VII. Examples

To illustrate the capabilities of the two routines, we consider three cases and the circuit decompositions for two of these.

Both programs are applied to the bridged-T equalizer circuit shown in Fig. 5(a); the response of the equalizer from 50 to 300 MHz is computed. In the SPEEDY routine, before the interconnection file can be constructed, the circuit has to be broken into 2-port blocks; this is shown in Fig. 5(b). For the GENERAL program, the circuit is broken into a cascade-load connection as shown in Fig. 5(c). By way of comparison, for five frequencies GENERAL took 0.967 s on a UNIVAC 1108 for a cost of about 19¢, while SPEEDY took 0.66 s on a Sigma 7 computer for a cost of about 10¢.

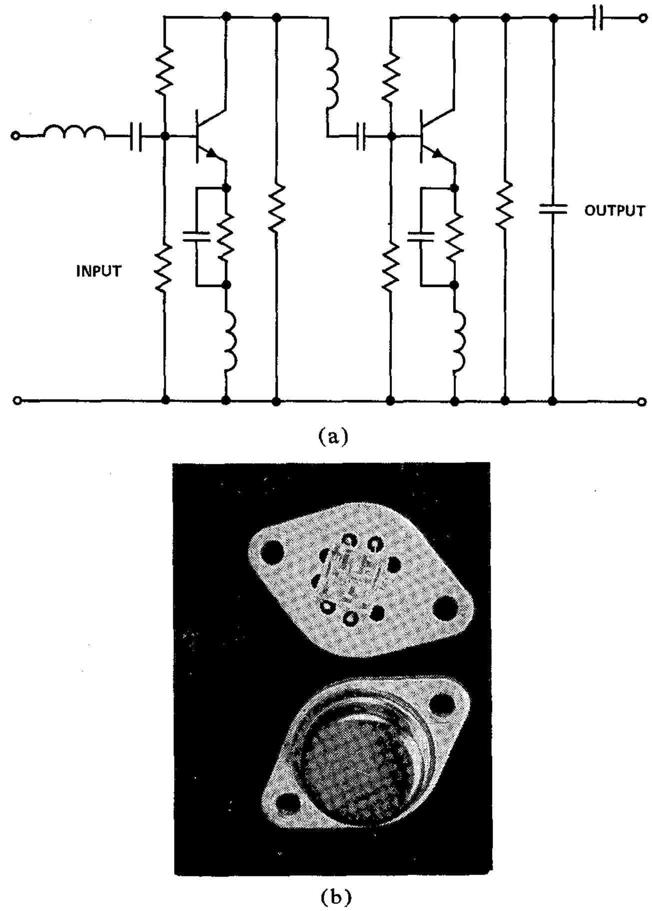


Fig. 6. (a) The RF equivalent circuit of the two-stage (20 dB gain) 20- to 100-MHz amplifier. (b) Picture of the circuit.

SPEEDY has been applied to many active and integrated 2-ports; for example, a typical active microcircuit, as shown in Fig. 6(a), has been designed by SPEEDY, resulting in the thin-film circuit shown in Fig. 6(b). However, Fig. 3(a) shows a circuit to which it cannot be applied. Thus, this amplifier was decomposed as previously discussed and shown in Fig. 3(b) for the application of GENERAL.

VIII. Discussion

By way of example, through two scattering matrix routines, we have developed here the idea of having two types programs available for the analysis of active networks. One, such as SPEEDY, is for handling very rapidly the normal day-to-day structure; in our case those 2-ports constructed by using standard 2-port interconnections. The other, such as GENERAL, is for handling all cases, although because of its complexity, it will undoubtedly handle them at higher cost and with longer computation time.

This specific example of implementing the philosophy of two complementary programs has been developed through the generality of the concept of cascade loading. Although various versions of the cascade-loading equation (3) have been around since the 1940's [19, p. 151], [22], we believe its generality in handling any connection [18] has not been recognized in the previous construction of scattering matrix routines. In particular, this generality has led us to the new formulation, as expressed in (10d), for classical 2-port connections. And this has significantly contributed to the speed of computation of SPEEDY.

Of course, as with most analysis routines, other subroutines can be easily added, for example, to evaluate sensitivities or to perform optimizations. In actual fact such an optimization subroutine is presently part of SPEEDY, though we have not discussed it here due to space limitations. It may, however, be worth mentioning that SPEEDY investigates stability by calculating the following stability factor k which, for unconditional stability, should satisfy [17, p. 6–4].

$$k = \frac{1 + |s_{11}s_{22} - s_{12}s_{21}|^2 - |s_{11}|^2 - |s_{22}|^2}{2|s_{12}s_{21}|} > 1$$

with

$$|s_{11}| < 1 \qquad |s_{22}| < 1 \tag{12a}$$

and

$$1 + |s_{11}|^2 - |s_{22}|^2 - |s_{11}s_{22} - s_{12}s_{21}|^2 > 0. \tag{12b}$$

If the circuit is potentially unstable, the program will furnish the graphical information on the regions of terminations that will maintain stable operation.

There are usually several choices of circuit decomposition which the user can, as a consequence, turn to his advantage. In particular, by a judicious choice of circuit interconnections, he may be able to cut computation time. For example, if a large number of active elements are present, it may be preferable to extract them several at a time in separate cascade sections, rather than all at one time as a load. However, experimentation has shown that improvements in calculation time are not extensive. What may be worth considering for the future is an automation of the decomposition steps themselves; this would seem relatively straightforward, but the cost has not seemed worth the effort at this point. Both programs as relatively convenient for the user to insert input data. Upon obtaining a suitable network decomposition, which most often suggests itself through design considerations, and on numbering ports and elements, one simply reads in the circuit element the desired descriptions and port interconnections.

As has been seen, the programs are set up to take scattering matrix data directly. However, other descriptions can readily be handled. For example, GENERAL converts admittances directly while SPEEDY accepts *RLC*, transistor, etc., element values. Nevertheless, the main interest of having programs available to accept measured data has been preserved, and since some 2-port data come from measurements on 3-terminal networks, SPEEDY contains a subroutine for converting data on 3-terminal devices to the various 2-port descriptions. In this regard the reader should be aware that the accuracy of the computed parameters greatly depends on the accuracy of the measured parameters and also on the values of the four basic terms of the original 2-port matrix. Experimental data, however, have proven the method used to be extremely accurate.

It should be made clear that GENERAL will consider many more external ports than in the two examples used, for example, up to 30 ports. What should be clear from the examples is that even though only two external ports are present, many more internal ports may be necessary. Specifically, this is the case where 2-port connections are used by the designer, but for some reason the 2-port hybrid matrices do not add. In this latter situation GENERAL will proceed to the proper result, which cannot be obtained by SPEEDY.

It should be mentioned that several other scattering matrix-type programs are being developed almost simultaneously [23]–[25]. One of these [23] is based upon the transfer scattering matrix and should be of considerable use where cascade networks are under analysis.

Because SPEEDY is designed for economical and rapid analysis, it is worthwhile stating a typical cost. For the 20-to 100-MHz circuit in Fig. 6, SPEEDY required 1.2 of CPU time at a total cost of 18¢ using the conversational time-shared mode of the Tymshare Sigma 7 computer for calculations at five frequencies (i.e., five passes through the frequency loop of Fig. 4 in 20-MHz steps). It is interesting to note that a hundredfold increase in the number of iterations increased the CPU cost only by a factor of 20, thus making optimization an economically reasonable task. As mentioned in the bridged-T example above, a comparison of SPEEDY with GENERAL has been made. Considering the difference in computing power between the Sigma 7 and the UNIVAC 1108 [26], SPEEDY proved to be about six times faster than GENERAL. Comparisons of SPEEDY with other programs, especially commercially available ones, are difficult because most of these programs will only print out the computer charges in dollars, and usually the royalty fees are included in the charges. One fair comparison, however, was made with Tymshare's MICAP using the same computer. After trying different circuit types, SPEEDY proved to be 20 to 50 percent faster with a particularly obvious advantage when active feedback circuits were evaluated. Some indirect comparisons with other commercial programs on different computers showed cost differentials often as high as 10/1 in favor of SPEEDY. This cost differential generally increased as the number of iterations required increased.

Everything, whole world's	Uncertainty
Good-evil, only	True certainties
Like shadow's body	Tightly appends
Ends judgement depends	All deeds [1, p. 8].

REFERENCES

[1] S. Pramoj, *Interpretive Translations of Thai Poets.* SEATO, Bangkok, 1965 (the second and third poems presented are retranslations).

[2] H. J. Carlin, "The scattering matrix in network theory," *IRE Trans. Circuit Theory*, vol. CT-3, June 1956, pp. 88–97.

[3] V. Belevitch, "Elementary applications of the scattering formalism in network design," *IRE Trans. Circuit Theory*, vol. CT-3, June 1956, pp. 97–104.

[4] Y. Oono and K. Yasuura, "Synthesis of finite passive 2n-terminal networks with prescribed scattering matrices," *Memoirs Faculty Eng.*, Kyushu University, Fukuoka, Japan, vol. 14, May 1954, pp. 125–177.

[5] R. W. Newcomb, *Linear Multiport Synthesis.* McGraw-Hill, 1966.

[6] L. Besser, "Combine *S* parameters with time sharing," *Electron. Des.*, Aug. 1, 1968.

[7] V. A. Monaco and P. Tiberio, "A method for automatic scattering matrix computation of a lumped, linear network," *Alta Freq.*, vol. 38, Nov. 1969, pp. 906–912.
[8] M. N. Parker, "DIPNET: A general distributed parameter network analysis program," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-17, Aug. 1969, pp. 495–505.
[9] P. L. Green, "General purpose programs for the frequency domain analysis of microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-17, Aug. 1969, pp. 506–514.
[10] P. E. Green, M. K. McPhun, M. A. Murray-Lasso, and A. E. Smoll, "Automatic general-purpose microwave circuit analysis programs," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-17, Aug. 1969, pp. 527–533.
[11] G. Valle, "A program for linear microwave circuit analysis by scattering matrices," *Alta Freq.*, vol. 38, Feb. 1969, pp. 142–144.
[12] V. A. Monaco, and P. Tiberio, "Automatic scattering matrix computation of microwave circuits," *Alta Freq.*, vol. 39, Feb. 1970, pp. 59–64.
[13] Hewlett-Packard information sheets, 1969.
[14] Fairchild Semiconductor, "*S* parameters for microwave transistors," 1968.
[15] KMC Semiconductor, Design Aids, 1970.
[16] *Hewlett-Packard J.*, Feb. 1967.
[17] "*S*-parameters . . . circuit analysis and design," Hewlett-Packard application Note 95, Sept. 1968.
[18] B. D. O. Anderson and R. W. Newcomb, "Cascade connection for time-invariant *n*-port networks," *Proc. Inst. Elec. Eng.*, vol. 113, June 1966, pp. 970–974.
[19] C. G. Montgomery, R. H. Dicke, and E. M. Purcell, *Principles of Microwave Circuits.* New York: McGraw-Hill, 1948.
[20] L. P. Huelsman, *Circuits, Matrices, and Linear Vector Spaces.* New York: McGraw-Hill, 1963.
[21] E. A. Guillemin, *Synthesis of Passive Networks.* New York: Wiley, 1957.
[22] M. R. Leibowitz, "Flow graph determination of the over-all scattering matrix of joined multiports," *Proc. IRE* (Corresp.), vol. 50, Sept. 1962, p. 1982.
[23] T. N. Trick, and J. Vlach, "Computer-aided design of broadband amplifiers with complex loads," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-18, Sept. 1970, pp. 541–547.
[24] V. A. Monaco and P. Tiberio, "A computer program for circuit analysis from d.c. to microwave using scattering parameters," in *Dig. Tech. Papers, 1970 IEEE Int. Symp. Circuit Theory*, pp. 119–120.
[25] P. Penfield, Jr., "MARTHA user's manual," M.I.T. Electrodynamics Memo 6, Sept. 21, 1970; Memo 12, Nov. 13, 1970.
[26] G. S. Pan, "Selection of time-sharing computer services," *Telecommunications*, May 1971, p. 43.

Existence of Solutions for the Equations of Transistor–Resistor–Voltage Source Networks

IRWIN W. SANDBERG, MEMBER, IEEE, AND ALAN N. WILLSON, JR., MEMBER, IEEE

Abstract—We consider the dc equations of nonlinear networks containing resistors, independent voltage sources (independent current sources are excluded), and certain types of nonlinear devices (such as Ebers–Moll-modeled transistors and diodes) that possess a certain property closely related to passivity. It is proved for the first time that the equations always possess at least one solution. This result complements some of the writers' previous work in which attention was not focused on networks containing only independent sources of the voltage type. In fact it was shown by simple examples given earlier that there exist transistor networks (containing ideal independent current sources) for which the network equations have no solution.

Here we also complete a study of conditions under which it is possible to carry out certain implicit numerical integration algorithms for the computation of the transient response of an important class of nonlinear networks containing transistors and diodes. We in fact prove that the assumption of passivity for the transistors and diodes implies that it is always possible to carry out the algorithms (in the sense that for any value of the step size there is always at least one solution of a certain key set of nonlinear equations).

Manuscript received March 10, 1971; revised June 11, 1971. This paper was presented at the 1970 IEEE International Symposium on Circuit Theory, Atlanta, Ga.
The authors are with Bell Telephone Laboratories, Inc., Murray Hill, N. J. 07974.

I. Introduction

This paper is concerned primarily with networks containing only transistors, diodes, resistors, and independent sources. We shall, however, focus attention on the more general network of Fig. 1 in order to emphasize that our results apply also to a much larger class of networks. It is, of course, clear that any network containing only transistors, diodes, resistors, and independent sources can be viewed as a network of the type shown in Fig. 1, with the nonlinear n-port containing the transistors and diodes, and the linear n-port containing the resistors and independent sources.

The results of this paper differ considerably from those of the writers' earlier related papers (see, for example, [1], [2]). In particular, here the emphasis is on 1) special properties of networks containing only independent sources of the voltage type, and 2) the implications of certain passivity, and passivity-like conditions.

The following notation is used throughout the paper. For each positive integer n we denote by E^n the n-dimensional