NONLISA: Nonlinear Network Simulation and Analysis Program

Toru Tsuda
Takuhito Kojima
Shinji Goto and
Toshihiko Nakamura

Fujitsu Limited
Kawasaki, Japan

1. Summary

Many programs for analyzing electronic circuits including nonlinear elements have been reported. It became difficult to evaluate circuits such as ICs by the breadboard method, because this method is not always suitable for simulation of high speed and high density circuits. Moreover lengthy experiments are involved. In the case of circuits composed of discrete parts, it was considered that many manual experiments could be replaced by computer aided analysis. We developed a general nonlinear network analysis program NONLISA to enable the use of medium scale computer system FACOM 270-30 for scientific and technical applications. This program is aimed mainly at the capability of circuit designers' easy treating and extending system functions without many modifications. Functions of NONLISA are DC analysis, transient analysis, sensitivity analysis, worst case analysis, Monte Carlo Simulation etc. for networks including nonlinear elements.

This paper deals with the structure and the functions of NONLISA.

2. Introduction

Electronic circuit analysis can be processed better by computer from among all the processes involved in electronic circuit design. Therefore computer aided electronic circuit analysis contributes towards considerably reduction in total design and experiment time. Many circuit analysis programs with nonlinear elements and their effectiveness have been reported in the past. We also felt the necessity of such analysis program for not only ICs, but circuits with many components and have developed NONLISA (Nonlinear network simulation and analysis program).

Main features of NONLISA are as follows:

Main features:

(1) Easy selection of dialogue form process or batch process.

(2) Easy registration and reference of devices.

(3) Prepared typical signal sources.

(4) Easy preparation of graphs obtained in transient analysis, input-output characteristics calculations and Monte Carlo simulation.

(5) Easy expansion and modification of the program system: this program is distinctly divided into the two, supervisory and analysis subprograms.

3. Program Structure

NONLISA consists of one supervisory program and several analysis and operation subprograms which are controlled by supervisory program, as in Fig. 1.

The supervisory program and network data is always stored on the core memory. Other analysis and operation programs are normally stored on the drum. They are loaded on the core memory overlay area by the supervisory program when instructed to do so by keyboard or card.

All the programs normally stored on the drum are divided into segments in accordance with each analysis. As mentioned above programs are modular in structure. Therefore, if interface between each subprogram is not changed, it is possible to level up and extend the system with modification and addition of subprograms only. Supervisory program consists of subroutines such as check program, registration program, command translation program. Therefore, it is also possible to level up and extend supervisory program in the same way. This method has to be adopted because all the programs can not be stored on FACOM 270-30 core memory which has a limited capacity of 65 KW (16 b/W). Moreover, it is considered that the time required for the program to be loaded onto the core memory from the drum in analysis operations does not in any give rise to undue problems to users. Both inner and outer drums have a maximum of 262 KW capacity.

FORTRAN is not considered to be particularly suitable for efficient core utilization. However it was selected to simplify application of this program to different types of computers with minor modification only. A total of about

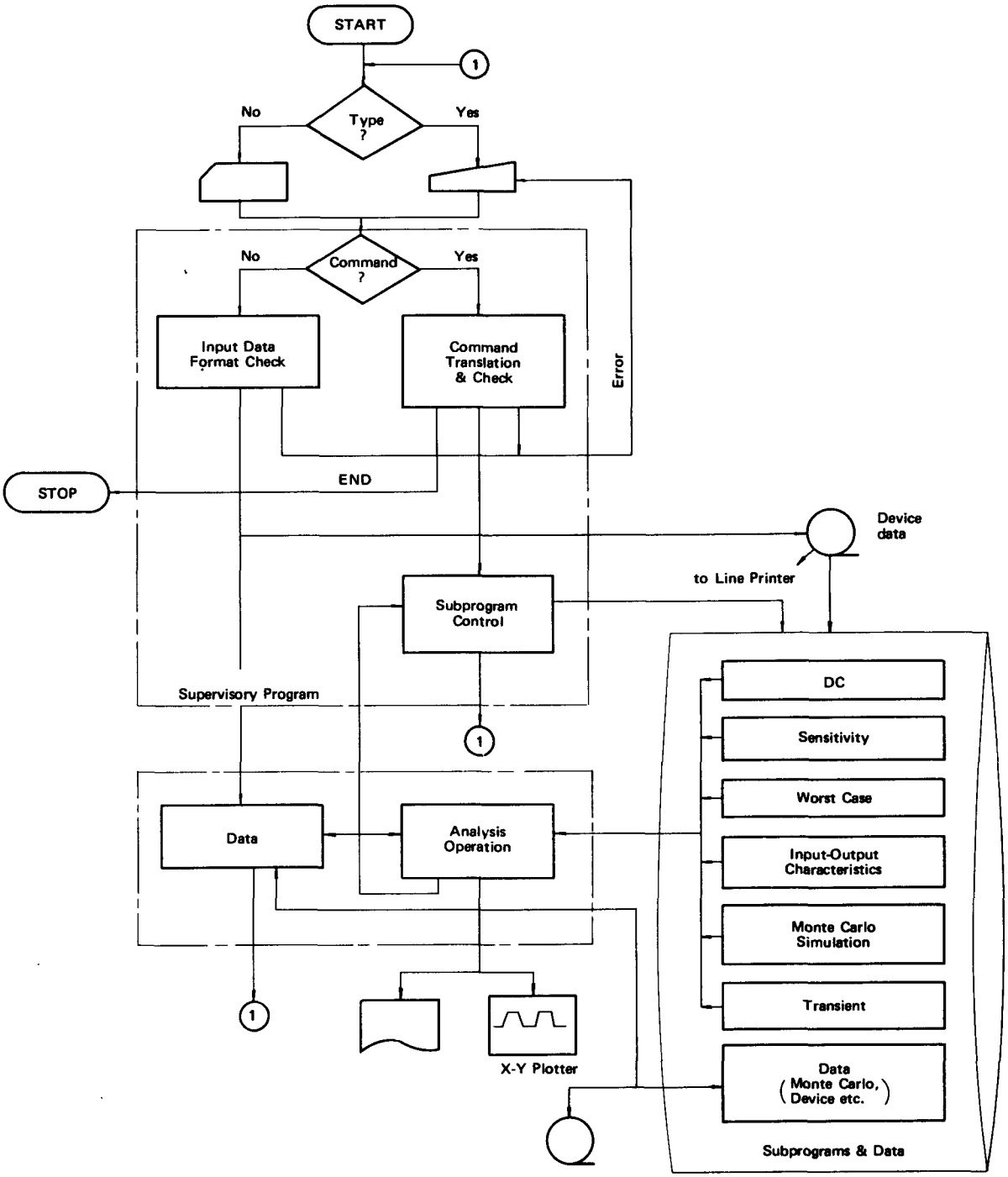


Fig. 1 The Program Structure of NONLISA

2,800 statements are used for the program. Details of this figure are shown in Table 1.

Table 1 Number of FORTRAN statements

Subprogram Name	Number of Statements
Supervisory prog.	450
Input data editting prog.	400
Output data editting prog.	290
Network equation generating prog.	160
DC analysis prog.	550
Transient analysis prog.	170
Sensitivity analysis prog.	120
Worst case analysis prog.	150
Monte Carlo simulation prog.	170
Input-output characteristic calculation prog.	80
Registration prog.	230
Total	2770

NONLISA required approximately 125 KW in length for these statements and data. These are divided into the two, core and drum. Details of this figure is shown in Table 2. In this table, 10 KW of program (overlay) means the maximum words of overlay area occupied by the program which is normally stored in the drum as mentioned previously.

Table 2 Occupied Memory Area of NONLISA

Memory	Use	Words (KW)
Core	Program (fixed)	13
	Program (overlay)	10
	Data	22
Drum	Program	36
	Data	54

The following analysis and operation programs are stored on the drum.

* DC analysis program.
* Sensitivity analysis program.
* Worst Case Analysis Program.
* Monte Carlo Simulation Program.
* Program for calculation of Input-Output Characteristics.
* Transient Analysis Program.

These are described in detail in chapter 6. Supervisory program operates only one of them at a time because from our experience in network design all of them are not always required simultaneously. However in Monte Carlo simulation, considerable data is required. Therefore in this type of simulation data parts are stored on the drum.

The user is able to select and use these analysis and operation programs in not only batch mode but also in dialogue mode with the keyboard.

4. Network for Analysis

NONLISA has the limitation as shown in Table 3 for the network to be analyzed.

Table 3 Limitation for Network Analysis

Limiting Factor	Limitation
Node	Max. 50
Branch	Max. 100
Input signal	Max. 5
Table	Max. 30 (section : max. 20)
Device registration	Transistor : max. 100 Diode : max. 100
Nonlinearity of device	One to one voltage to current, voltage to charge, or flux to current characteristics

5. The Features of the Supervisory Program

The supervisory program is always stored on the core memory. It controls all analysis and operation programs.

Its functions can be summarized as follows.

* Translation of command and input data.
* Loading of analysis and operation programs from the drum to the core memory in accordance with the user's commands.
* Checking command sequence, input date format, etc.
* Processing in case of errors.
* Interface among analysis or operation subprograms.
* Control of output data.

The number of subprograms controlled by supervisory program is limited to maximum number of drum segments. But the number of segments being too large the number of subprograms is not necessarily effected.

6. Method of Analysis

Method of analysis is based on the papers by J. Katzenelson. It is as follows.

(1) Network description (1), (2), (4)

It is necessary to arrange network data in the form convenient to analyze, to select a tree and to make a network matrix, before analyzing. State variable method is used in transient analysis. In this method DC analysis can be used. Therefore, the same network description is adopted in DC and transient analysis.

Network is expressed by dividing branches into trees and links based on graph theory. The following normal tree is chosen as a tree. A normal tree is a tree which contains all voltage sources as tree branches all current sources as links and as many capacitive branches and inductive links as possible. Network equations are expressed by a cutset matrix containing only one tree branch in each cutset. That is, each branch is arranged in the following priority sequence and a tree branch is chosen.

(a) voltage sources, signal voltage sources and dependent voltage sources.
(b) capacitors
(c) resistors
(d) inductors
(e) current sources, signal current sources and dependent current sources.

These equations are based on the concept that total sum of current in each cutset equals zero. They are solved by inverse matrix operation of admittance matrix.

(2) DC Analysis (3)

In the case of nonlinear network initial condition of the network is given. Then the network matrix is solved iteratively until all assumed sources that satisfy matrix equations coincide with the corresponding real sources. Branch voltages at the time of coincidence are the ones required. Nonlinear characteristics are expressed in piecewise linear form. Inverse matrix calculation is necessary for every segment of nonlinear conductances. As the difference between adjacent conductance segments is small, inverse matrix can be calculated effectively by Kron's method.

This method shall be described in further detail. Generally, the cutset equation of a network can be expressed as follows.

$$\mathbb{G}\begin{pmatrix}\mathbb{e}\\ \mathbb{E}\end{pmatrix} = \mathbb{J} \qquad (1)$$

where $\mathbb{G}$ is a matrix corresponding to conductance characteristic, $\begin{pmatrix}\mathbb{e}\\ \mathbb{E}\end{pmatrix}$ is a vector whose components are the branch voltage vector $\mathbb{e}$ and voltage source vector $\mathbb{E}$, and $\mathbb{J}$ is a current source vector.

The following method is used to obtain the solution i.e. $\mathbb{e}$. First, any $\mathbb{e}$ is selected. Let it be $\mathbb{e}_0$. $\mathbb{J}_0$ corresponding to $\mathbb{e}_0$ is calculated from equation (1). That is,

$$\mathbb{G}\begin{pmatrix}\mathbb{e}_0\\ \mathbb{E}\end{pmatrix} = \mathbb{J}_0 . \qquad (2)$$

Normally $\mathbb{J}_0$ and $\mathbb{J}$ are not equal. This is because $\mathbb{e}_0$ is selected arbitrarily. Therefore $\mathbb{e}_0$ cannot be normally considered to be the solution.

Then $\mathbb{e}_1$ which is closer to the actual solution is selected. To determine whether it is the desired solution, the following method is used. Such operations are repeated until all the assumed sources satisfy equation (1).

Closer branch voltages are selected as follows.

$\mathbb{e}_\lambda$, which satisfies the following equation for $0 \leqq \lambda \leqq 1$, is considered

$$\mathbb{G}\begin{pmatrix}\mathbb{e}_\lambda\\ \mathbb{E}\end{pmatrix} = \mathbb{J}_0 + \lambda(\mathbb{J}-\mathbb{J}_0) . \qquad (3)$$

In this equation,
for $\lambda = 0$, $\mathbb{e}_\lambda = \mathbb{e}_{\lambda 0}$; which is the starting point.
and
for $\lambda = 1$, $\mathbb{e}_\lambda = \mathbb{e}$; which is the solution of (1).

Selection of a value for λ which is closer to the actual solution at each step until $\lambda = 1$ is obtained is carried out as follows.

Let $\mathbb{G}_1$ denote the conductance matrix in the first linear region. In order to find the second point $\mathbb{e}_{\lambda 1}$ on the boundary of the region which includes initial point $\mathbb{e}_0$, the largest λ_1 is found so that $\mathbb{e}_0 + \lambda_1 \mathbb{G}_1^{-1}(\mathbb{J} - \mathbb{J}_0)$ is in the same linear region.
That is, $\mathbb{e}_{\lambda 1} = \mathbb{e}_0 + \lambda_1 \mathbb{G}_1^{-1}(\mathbb{J} - \mathbb{J}_0) \equiv \mathbb{e}_0 + \lambda_1 \Delta \mathbb{e}_0$.
Next, third point $\mathbb{e}_{\lambda 2}$ is calculated from $\mathbb{e}_{\lambda 1}$ and $\mathbb{J}_1$ which corresponds to $\mathbb{e}_{\lambda 1}$ in the same way. This process is repeated until $\lambda = 1$, that is until $\mathbb{e}$ reaches the solution.

Convergence of this process has been proved under the following conditions,

(a) network consists of a finite number of branches.

(b) characteristics of each element are continuous, piecewise linear and monotonic increasing function, and has a finite number of breakpoints at any interval.

This process is shown in Fig. 2.

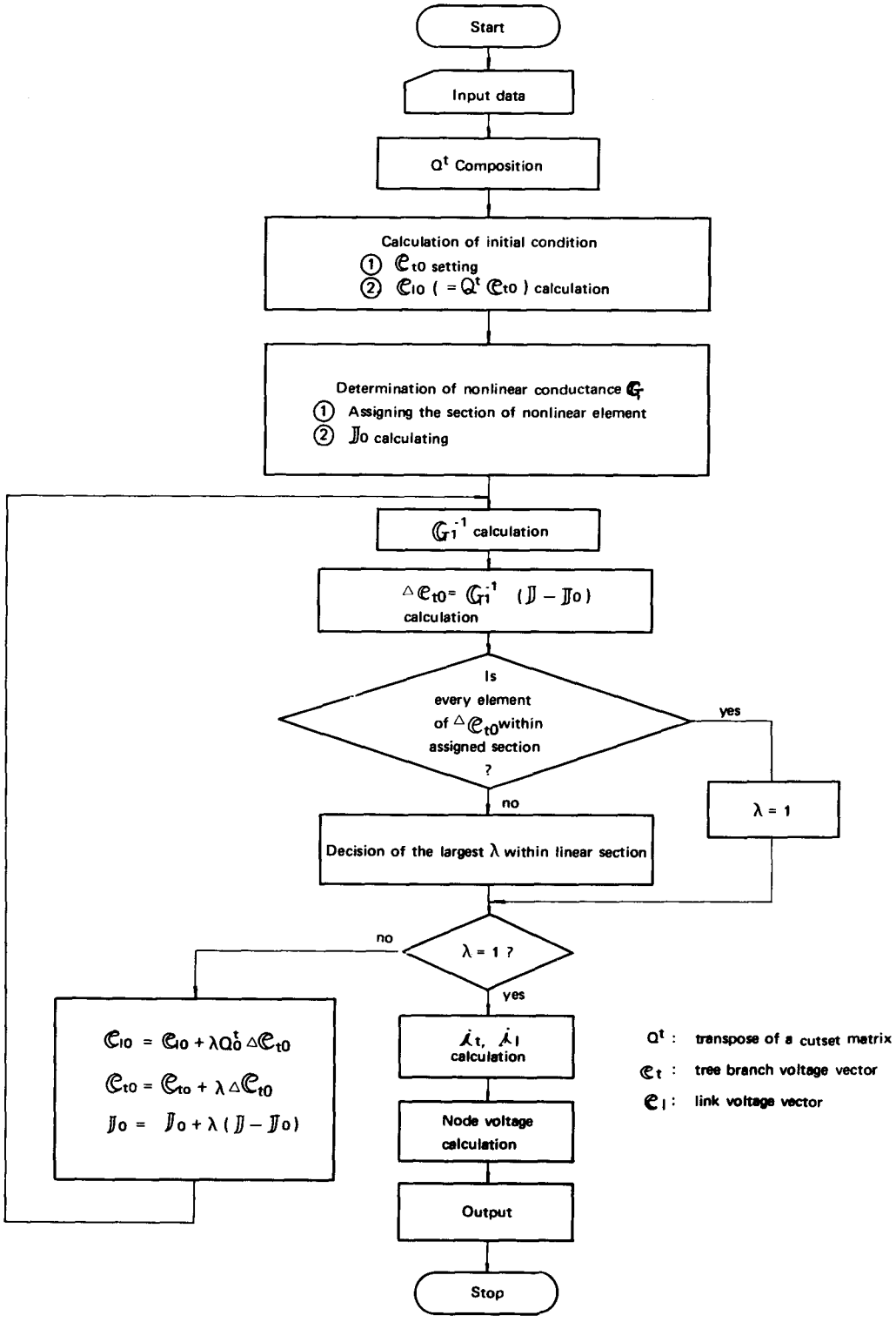


Fig. 2 Flow Chart of DC Analysis

(3) Transient Analysis Method (2), (4)

Generally, electric network can be expressed by the following state equation, in which variables are state variables. They correspond to the stored energies such as capacitor voltages and inductor currents.

$$\dot{\mathbb{X}} = \mathbb{A}\mathbb{X} + \mathbb{B}\mathbb{U}$$

where $\mathbb{X}$ is state variable vector, $\mathbb{U}$ is input vector, and $\mathbb{A}$ and $\mathbb{B}$ are corresponding constant vectors. It is considered that the state variable method is suitable for obtaining transient solution by iterative computation. That is, network can be expressed by the three one-element-kind networks i.e. R, L and C network and state variables for a short time. As one-element-kind networks can be analyzed with DC analysis method, transient analysis becomes simple.

Next, this method shall be taken up in further detail. Network η is divided into three one-element-kind networks as shown in Fig. 3.

Subnetwork	E	C	R	L	J
η_C	S	*	0	0	0
η_R	S	S	*	0	0
η_L	S	S	S	*	0

Fig. 3 Three One-element-kind Networks

In Fig. 3, S or O signifies replacement of each branch of E, C, R, L or J by a short circuit or an open circuit respectively.

Next, equivalent voltage and current sources are connected at a point that corresponds to S and O. The values of these equivalent sources are sources and capacitor branch voltage, inductor branch current and resistor branch voltage and current which are sources and state variables at that time. As Kirchhoff's law is valid only by introducing equivalent sources to such subnetworks, η can be replaced by η_c, η_R, η_L and equivalent sources. If equivalent sources at that time are given, element voltages and currents of each of the subnetworks can be obtained in the same way as in DC analysis. As differential values of state variables can be achieved by solving subnetworks, numerical integration is required in order to obtain the state variables. These are used as state variables in the next computation. By repeating such process during desired time, transient solution can be obtained.

Flow diagram of this transient analysis is shown in Fig. 4.

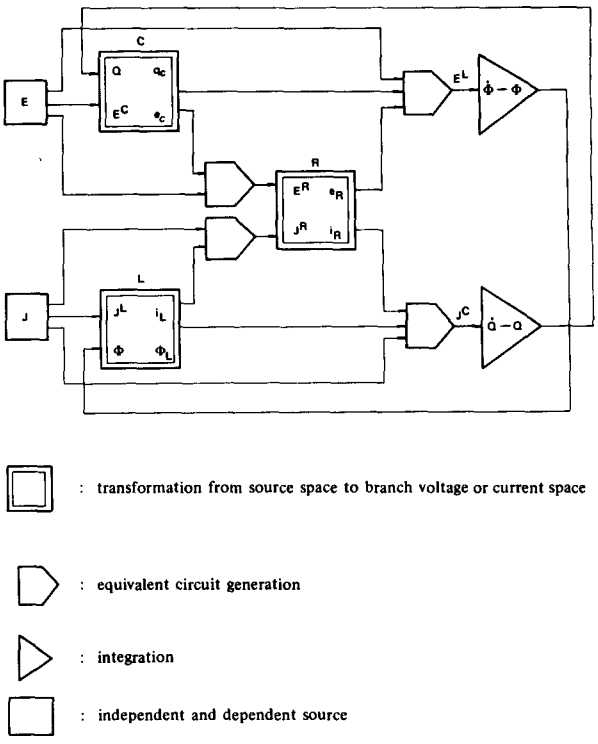


Fig. 4 Flow Diagram of Transient Analysis

(4) Other operation programs

Other operation programs that utilize results obtained by DC analysis are described next.

(a) Sensitivity Analysis Program

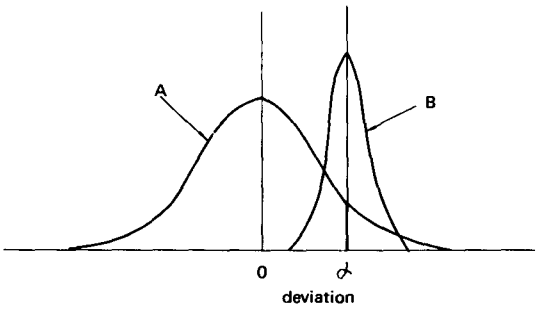
DC nominal solutions are obtained by the method described in 6 (1) and (2). DC solution for the desired outputs for the change of an assigned element value can be obtained with sensitivity analysis program. In NONLISA, it is allowed to assign arbitrary or fixed deviation of source or element values.

The following method is adopted. Sensitivity solution is obtained by subjecting the network with parameter modified with given deviation to DC analysis.

(b) Worst Case Analysis Program

This is the program by which DC worst case value of assigned output is calculated with change of sources or parameters of elements. The worst case is determined by choosing the direction of parameter deviation which caused the same directional change of given output. After determining the element parameter by its polarity, DC worst case solution is obtained by means of DC analysis

program. It is also considered to be able to use easily the deviation with correlation between deviations of element parameters such as ICs, as shown in Fig. 5.



A: distribution curve of one element.
B: distribution curve of another element.
α: one deviation in A.

Fig. 5 Relation between deviations with correlation

(c) Monte Carlo Simulation Program

This program is as follows. First, a set of values of all sources and elements in the network are selected within assigned deviations of the desired output by using normal random numbers. Second, DC analysis is solved for the set of parameter values. This process is repeated up to the assigned iterative number. As a result, distribution of DC characteristic such as output voltage is obtained. Ordinarily any iterative number of less than 31,500 is chosen in NONLISA. Use of deviation with correlation between element parameters as well as in worst case analysis is allowed.

It is possible to print out the result of this analysis in the form of histogram.

In this simulation, every operation is performed on the core memory, but the results are stored on the drum as considerable data are required.

(d) Input-output Characteristics Calculation Program

This is the program which calculates output responses and derivatives of variation in response to the change of assigned parameter. It is provided to draw results of this calculation on the X–Y plotter.

7. Registration of Device

The user may input the nonlinear devices such as transistors and diodes into the computer in a tabular form. But it is troublesome to arrange data of frequently used diodes and transistors in every calculation. Therefore, it is provided for users to register device data into the magnetic tape and to use them with registered name.

Nonlinear resistors is expressed in piecewise linear approximation as shown in Fig. 6. Fig. 7 shows its equivalent circuit. Each section is expressed as a set of voltage and current and all sections are expressed in tabular form as follows.

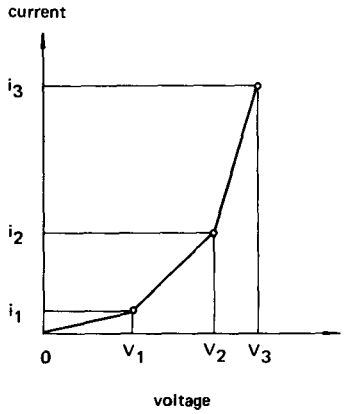


Fig. 6 Piecewise Linear Approximation of Nonlinear Resistor

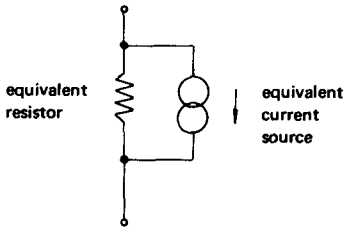


Fig. 7 Equivalent Circuit of Nonlinear Resistor

* TABLE n
△△△XXX,△ △ △ XXX,- - -
where △△△ and XXX denotes voltage and current respectively.

Nonlinear capacitor and inductor are also expressed in the same way.

Diodes are considered to be nonlinear resistors.

Transistors can be expressed in Ebers-Moll model in Fig. 8. But it is necessary to input each element data separately for other models such as the model including parasitic elements. Fig. 9 shows an example of device assignment. When a registered name of an device is designated, the tables corresponding to it are assigned in the program. For example, when registered name is designated, a particular table for diode or two tables for transistor are assigned.

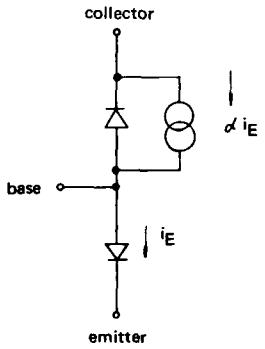


Fig. 8 Ebers-Moll Equivalent Circuit of NPN Transistor

* Nonlinear Resistor :
  NR ○ ○ ℓ - m = TABLE k
* Diode :
  D ○ ○ ○ ℓ - m = XXXXXXX
* Transistor :
  Q ○ ○ ℓ - m - n = XXXXXXX α or β

where
○ : device number
l, m, n : node number
k : table number
X ···· X : registered device name

Fig. 9 Assignment of Device

These registered data are filed into the magnetic tape. These data must be transferred to the drum by the user prior to execution.

The user is able to register easily these device data into the MT. He can also print out the data from MT on the line printer by means of specific commands. This can be done easily in the dialogue form with the computer.

8. Main Features of Other Subprograms

(1) Preparation of Signal Sources Information

In transient analysis, the user can input arbitrary signal sources in tabular form. The following signal source functions defined in time $t > 0$, are prepared for the user's convenience.

(a) Sine wave
(b) Amplitude modulated sine wave
(c) Exponential function wave
(d) Error function wave
(e) Periodical trapezoid wave

The user can easily input the desired signal sources by assigning function name and its parameters.

(2) Graphical Expression of Analysis Results

It is convenient to express analysis results with graph as much as possible. Graphs such as input-output characteristics and output waveforms can be drawn with the X–Y plotter, and figures such as histogram can be done with the line printer.

Fig. 15 and Fig. 16 are examples of input-output characteristic drawn with the X–Y plotter, and histogram of the result of Monte Carlo Simulation drawn with the line printer respectively. Unless specifically designated, the results of analysis are not drawn.

(3) Checking Function

Checking function of the user's error such as command format error, data format error is prepared for preventing wasteful calculation.

Errors are classified into serious and slight error according to their extent. In the case of serious error, the user must correct wrong input data and repeat the calculation from the beginning. In the case of slight error, the user can continue to calculate after correction in dialogue form with the keyboard or after correction of card. The following ones can be considered to be serious errors.

* assigning node or branches in excess of network scale.
* incomplete network connection.
* no registration of assigned device.
* assigning prohibited sequence of the command. etc.

The following ones can be considered to be slight errors.

* error of input data format.
* error of command format.
* assigning wrong command name.
* assigning illegal output variable. etc.

NONLISA error messages are given in table 4.

Table 4 Error Message List

Error No.	Error Message
01	Not permitted element
02	Table n is not registered
03	Signal n is not registered
04	Computation time is not appointed
05	Output variables are not appointed
06	Illegal command statement
07	Illegal output variable
08	Illegal parameter variable
1001	Network scale over
1002	Device data format error
1003	Device data sequence error
1004	No device name
1005	Network composition error
1006	Cannot be solved
1007	Illegal BFUNC parameter
1008	Illegal command parameter
1009	Illegal table list

01 ~ 08 : slight error
1001 ~ 1009 : serious error

(4) Output of Auxiliary Data

Auxiliary data such as calculation accuracy, graphic composition of network, iterative frequency of calculation up to solution, are in some case necessary for the user or programmers. These can be output on the keyboard by commands from it.

The sum of each cutset current is considered as calculation accuracy. This sum is essentially zero, but normally it is not found to be zero because of calculation error. The user can output the maximum value of the sum on the keyboard. Iterative calculation frequency is the frequency in which calculation converges in the case of nonlinear network, that is, λ reaches 1 as mentioned previously.

9. Command List of NONLISA

NONLISA is the dialogue form program using commands as mentioned above. These commands are shown in table 5. Effort has been made to make the terminology as uniform with other programs of our company as possible.

Table 5 Command List

Command Statement	Explanation
* REGISTRATION	Device registration into MT and reference of registered device.
* INSTRUCTION	Output instruction of auxiliary information.
* DELEMENT	Device data transference from MT to Drum.
* ELEMENT	Input of circuit data.
* TABLE	Input of nonlinear device.
* SIGNAL	Input of provided signal source.
* TIME=	Step of calculation time in transient analysis.
* PTIME=	Step of print out time in transient analysis.
* TBEGIN	Starting time of print out in transient analysis.
* TEND	End time of calculation in transient analysis.
* OUTPUT	Assignment of output in analysis except for steady state solution.
* PARAMETER	Assignment of variable input in sensitivity, Monte-Carlo or worst case analysis.
* STEADY	Execution of DC analysis.
* IOCHARACT	Execution of input-output characteristics.
* SENSE	Execution of sensitivity analysis.
* WORST	Execution of worst case analysis.
* MONTE	Execution of Monte Carlo simulation.
* TRANSIENT	Execution of transient analysis.
* END	End of calculation

10. An Example of Analysis

An example of analysis is shown in Fig. 10 ~ 17.

(1) Fig. 10 shows a circuit example which is a typical logic circuit TTL.

(2) Fig. 11 shows the user's coding for transient analysis of a circuit example including DC analysis.

(3) Fig. 12 shows element list of this circuit. Transistors are expressed with Ebers-Mill model in Fig. 8.

(4) Network composition based on graph theory is expressed in a cutset matrix. This matrix is shown in Fig. 13.

(5) Fig. 14 shows node voltages, branch voltages, currents and power dissipation obtained by DC analysis.

(6) Fig. 15 shows DC input-output characteristic drawn with the X–Y plotter.

(7) Fig. 16 shows a histogram of source current distribution calculated by Monte Carlo Simulation.

(8) Fig. 17 shows input and output waveforms drawn with the X–Y plotter.

11. Conclusion

NONLISA, a nonlinear network simulation and analysis program using state variable method and having the features mentioned above, has been described here. Analysis of numerous electronic circuits by this program has proved the usefulness of these features to the user. However, proper device model especially proper high speed transistor model has not been found yet. Realization of such a model will contribute to increasing the effectiveness of analysis program to a considerable extent.

12. Acknowledgment

The authors wish to thank K. Toyoda, Assistant Director of Switching System Division for the opportunity to publish this paper. They are also indebted to Mr. K. Uehara for his assistance in programming and Mr. T. Kojima of FUJITSU LABORATORIES LTD. for helpful discussion.

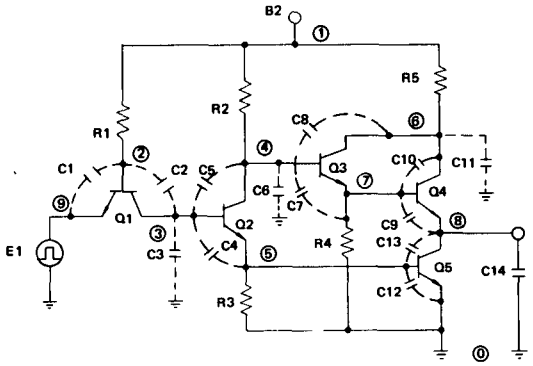
13. References

(1) S. Seshu and M. B. Reed, "Linear Graph and Electrical Networks", Reading, Mass.: Addison-Wesley, 1961.

(2) C. A. Desoer and J. Katzenelson, "Nonlinear RLC Networks", B.S.T.J., vol. 44, PP. 166-198 Jan. 1965.

(3) J. Katzenelson, "An Algorithm for Solving Nonlinear Resistor Networks", B.S.T.J., vol. 44, PP. 1605-1620 Oct. 1965.

(4) J. Katzenelson, "AEDNET: A Simulator for Nonlinear Networks", Proc. IEEE, vol. 54, No. 11 PP. 1536-1552, Nov. 1966.



R1: 2.8KΩ	C1: 1.6PF	C10: 0.3PF
R2: 760Ω	C2: 2.3PF	C11: 3.0PF
R3: 470Ω	C3: 1.3PF	C12: 8.8PF
R4: 4KΩ	C4: 3.4PF	C13: 0.9PF
R5: 58Ω	C5: 0.4PF	C14: 15PF
Q1: 2SCXX	C6: 1.2PF	
Q2: 2SCXX	C7: 2.3PF	
Q3: 2SCXX	C8: 0.3PF	
Q4: 2SCXX	C9: 2.3PF	
Q5: 2SCXXXX		
E1: SIGNAL1		
B2: 5V		

Fig. 10 Circuit Example

```
***INPUT DATA ***
*INSTRUCTION
*ELEMENT
      R1 1-2=2.8(0.2)
      R2 1-4=0.76(0.2)
      R3 5-0=0.47(0.2)
      R4 7-0=4.0(0.2)
      R5 1-6=0.058(0.2)
      Q5 0-5-8=2SCXXXX
      Q4 8-7-6=2SCXX
      Q3 7-4-6=2SCXX
      Q2 5-3-4=2SCXX
      Q1 9-2-3=2SCXX
      C1 2-9=1.6
      C2 2-3=2.3
      C3 3-0=1.3
      C4 3-5=3.4
      C5 3-4=0.4
      C6 4-0=1.2
      C7 4-7=2.3
      C8 4-6=0.3
      C9 7-8=2.3
      C10 7-6=0.3
      C11 6-0=3.0
      C12 5-0=8.8
      C13 5-8=0.9
      C14 8-0=15
      E1 9-0=SIGNAL1
      B2 1-0=5(0.1)
*OUTPUT
      V8=GRAPH1
      V9=GRAPH1
*SIGNAL1
      0 0,10 0,15 3,30 3,35 0,50 0
*TIME=0.01
*PTIME=0.5
*TEND=50
*TRANSIENT
*END
```

Fig. 11 Coding for Transient Analysis

********** ELEMENT LIST **********

NO	NAME		NODE	VALUE	DEVIATION	DD	PARAMETER
1	R1		1- 2	0.28000E+01	0.20000	0	
2	R2		1- 4	0.76000E+00	0.20000	0	
3	R3		5- 0	0.47000E+00	0.20000	0	
4	R4		7- 0	0.40000E+01	0.20000	0	
5	R5		1- 6	0.58000E-01	0.20000	0	
6	Q5	E	5- 0	2SCXXXX	0.00000	0	
7	Q5	C	5- 8	2SCXXXX	0.00000	0	
8	Q5	J	8- 5	0.98592E+00*JS5	E 0.00000	0	
9	Q4	E	7- 8	2SCXX	0.00000	0	
10	Q4	C	7- 6	2SCXX	0.00000	0	
11	Q4	J	6- 7	0.98571E+00*JS4	E 0.00000	0	
12	Q3	E	4- 7	2SCXX	0.00000	0	
13	Q3	C	4- 6	2SCXX	0.00000	0	
14	Q3	J	6- 4	0.98571E+00*JS3	E 0.00000	0	
15	Q2	E	3- 5	2SCXX	0.00000	0	
16	Q2	C	3- 4	2SCXX	0.00000	0	
17	Q2	J	4- 3	0.98571E+00*JS2	E 0.00000	0	
18	Q1	E	2- 9	2SCXX	0.00000	0	
19	Q1	C	2- 3	2SCXX	0.00000	0	
20	Q1	J	3- 2	0.98571E+00*JS1	E 0.00000	0	
21	C1		2- 9	0.14000E+01	0.00000	0	
22	C2		2- 3	0.23000E+01	0.00000	0	
23	C3		3- 0	0.13000E+01	0.00000	0	
24	C4		3- 5	0.34000E+01	0.00000	0	
25	C5		3- 4	0.40000E+00	0.00000	0	
26	C6		4- 0	0.12000E+01	0.00000	0	
27	C7		4- 7	0.23000E+01	0.00000	0	
28	C8		4- 6	0.30000E+00	0.00000	0	
29	C9		7- 8	0.23000E+01	0.00000	0	
30	C10		7- 6	0.30000E+00	0.00000	0	
31	C11		6- 0	0.30000E+01	0.00000	0	
32	C12		5- 0	0.88000E+01	0.00000	0	
33	C13		5- 8	0.90000E+00	0.00000	0	
34	C14		8- 0	0.15000E+02	0.00000	0	
35	E1		9- 0	SIGNAL 1	0.00000	0	
36	B2		1- 0	0.50000E+01	0.10000	0	

********** SIGNAL LIST **********

SIGNAL 1

SEG	TIME	V/I
1	0.00000E+00	0.00000E+00
2	0.10000E+02	0.00000E+00
3	0.15000E+02	0.50000E+01
4	0.30000E+02	0.50000E+01
5	0.35000E+02	0.00000E+00
6	0.50000E+02	0.00000E+00

Fig. 12 Element List

********** NETWORK COMPOSITION **********

*TREE BRANCH= 33 34 21 22 24 23 27 28 29
* LINK = 23 26 30 31 32 35 36 1 2 3 4 5 6 7 9 10 12 13 15 16 18 19 8 11 14 17 20

** CUT SET MATRIX **

	23	26	30	31	32	33	34	1	2	3	4	5	6	7	9	10	12	13	15	16	18	19	8	11	14	17	20
35	1	1	0	1	1	0	1	-1	1	1	-1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	1	1	0	1	1	0	1	-1	-1	1	1	-1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0
22	-1	-1	0	-1	-1	0	-1	0	1	-1	-1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	-1
24	0	0	0	0	-1	-1	0	0	0	-1	0	0	-1	-1	0	0	0	0	0	1	0	0	0	1	0	0	0
25	0	-1	0	-1	0	1	-1	0	1	0	-1	1	0	0	1	0	0	0	0	0	1	0	0	-1	0	0	-1
27	0	0	-1	0	0	1	-1	0	0	0	-1	0	0	1	0	-1	1	0	0	0	0	0	0	-1	1	0	0
28	0	0	1	-1	0	0	0	0	0	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	-1	0	0
29	0	0	0	0	0	0	1	-1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	-1	0	0	0

Fig. 13 Network Composition (Cutset Matrix)

********** STEADY STATE SOLUTION **********

**ERROR ESTIMATION VALUE Eps= 0.16212E-04

***NODE VOLTAGE SOLUTION

1	0.50000E+01	2	0.22899E+01
3	0.15118E+01	4	0.73937E+00
5	0.75140E+00	6	0.49948E+01
7	0.28096E+00	8	0.15052E+00
9	0.40000E+01		

***ELEMENT SOLUTION

NO	NAME		VOLT	CURR	VOLT*CURR
1	R1		0.27101E+01	0.96790E+00	0.26231E+01
2	R2		0.42606E+01	0.56061E+01	0.23885E+02
3	R3		0.75140E+00	0.15987E+01	0.12015E+01
4	R4		0.28096E+00	0.70241E-01	0.19735E-01
5	R5		0.51795E-02	0.89302E-01	0.46255E-03
6	Q5	E	0.75140E+00	0.49943E-01	0.37527E-01
7	Q5	C	0.60088E+00	0.49039E-01	0.29467E-01
8	Q5	J	-0.60088E+00	0.49240E-01	-0.29587E-01
9	Q4	E	0.13044E+00	0.20068E-01	0.26178E-02
10	Q4	C	-0.47139E+01	-0.47139E-06	0.22220E-05
11	Q4	J	0.47139E+01	0.19787E-01	0.93248E-01
12	Q3	E	0.45843E+00	0.70927E-01	0.32532E-01
13	Q3	C	-0.42555E+01	-0.42555E-06	0.18109E-05
14	Q3	J	0.42555E+01	0.66520E-01	0.29584E+00
15	Q2	E	0.76042E+00	0.65730E-01	0.49982E-01
16	Q2	C	0.77244E+00	0.87400E+00	0.67511E+00
17	Q2	J	-0.77244E+00	0.64791E-01	-0.50047E-01
18	Q1	E	-0.17101E+01	-0.17101E-06	0.29245E-06
19	Q1	C	0.77807E+00	0.96790E+00	0.75310E+00
20	Q1	J	-0.77807E+00	-0.16857E-06	0.13116E-06
21	B1		0.40000E+01	-0.17101E-06	-0.68405E-06
22	B2		0.50000E+01	-0.66693E+01	-0.33417E+02

Fig. 14 Results of DC Analysis

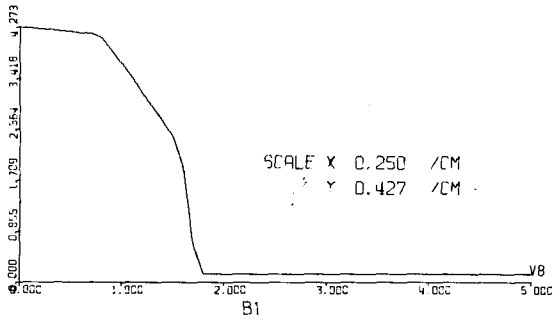


Fig. 15 DC Input-Output Characteristic

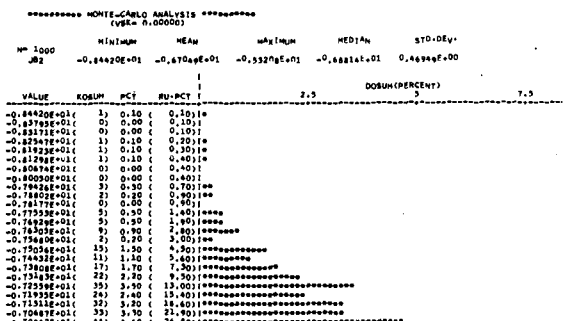


Fig. 16 Histogram of Source Current Distribution with Monte Carlo simulation

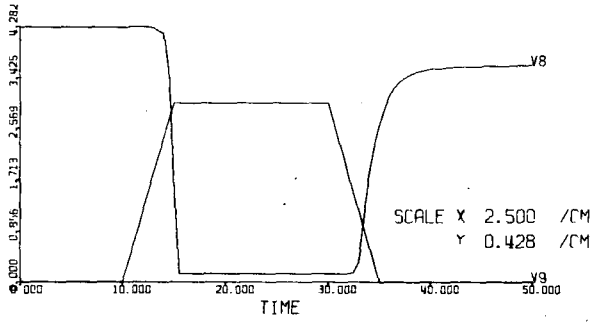


Fig. 17 Input and Output Waveforms