September, 1968 Status Report ESL-SR-363 Copy______

COMPUTER-AIDED ELECTRONIC CIRCUIT DESIGN

Status Report

December 1, 1967 - May 31, 1968

The preparation and publication of this report, including the research on which it is based, was sponsored under a grant to the Electronic Systems Laboratory, Massachusetts Institute of Technology, DSR Project No. 76152. This grant is being administered as part of the National Aeronautics and Space Administration Research Grant No. NsG-496 (Part). This report is published for information purposes only and does not represent recommendations or conclusions of the sponsoring agency. Reproduction in whole or in part is permitted for any purpose of the United States Government.

Approved by John E. Ward

John E. Ward
Deputy Director

Electronic Systems Laboratory
Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

Research in On-Line Circuit Design was pursued along a number of directions, centering on the effective use of present and projected on-line utilities in the design of electrical networks and systems. This work may be divided into two broad categories, research in the mathematical foundations of computer-oriented network and system analysis and the interactive features essential for the design of networks and systems. Progress has been made in the development of CIRCAL-II, and in particular on the development of the CIRCAL-II data structure; in a recursive approach for the computer analysis of nonlinear networks; in answering the question "when and how should a network be torn in order to reduce the necessary computational effort"; and in the development of LOTUS, an on-line program for the simulation of block-diagram systems.

CONTENTS

LIST OF FIGURES

A. INTRODUCTION

The main objective of this research is the effective use of present and projected on-line utilities in the design of electrical networks and systems. This includes research in the mathematical foundations of computer-oriented network and system analysis, and the interactive features essential for the design of networks and systems.

During the reporting period, the research activity of this group was divided into two main categories: networks where relationships are primarily implicit, and so-called block-diagram systems where relationships are explicit. In the case of networks, emphasis was placed in the development of an integrated on-line circuit-design system, CIRCAL-II, which is now operational. This evolutionary system and its predecessor, CIRCAL-I, are used as an experimental forum to test and incorporate research results on new computer-oriented network-simulation techniques. Theoretical work in the same area entailed continuation of research in the development of a recursive approach for the computer analysis of nonlinear networks and in the tearing of nonlinear networks. In the case of systems, an "equivalent" on-line simulator LOTUS-1 was developed, in which analog or digital systems are simulated as compositions of primitive functions and functionals.

B. CIRCAL-II

Professor M. L. Dertouzos
Mr. G. P. Jessel,
Research Assistant
Mr. J. R. Stinger
Research Assistant

The first version of CIRCAL-II has been implemented. It is now possible to: create a circuit, consisting of standard elements and nested structures; analyze it, have the results either plotted or printed; and from this data make changes in either the network topology or some parameter and then repeat analysis.

As shown in Fig. 1, the operations of CIRCAL-II may be collected into three groups :

1. Basic file system operations
2. Setup of a data structure and subsequent analysis
3. Output of requested data and modification of the circuit under investigation

In addition, all parameters relevant to the output phase of operation are stored in a "mode file". This permits the user to simply indicate incrementally any desired changes and then perform re-analysis, rather than having to specify each time the entire set of necessary parameters. A global diagnostic procedure is also provided which indicates to the CIRCAL-II user the type of error encountered (out of about 30 types), and whenever possible makes appropriate corrections.

The entire file system of CIRCAL-II is now in operation. It consists of a set of master files which contain the smaller files of CIRCAL-II. This has been done to accomodate the smaller files encountered in circuit design (20-50 words) with the minimum file size of 432 words required by CTSS. Five types of files have been specified,namely: circuits, nested structures, functions, functionals, and defined commands. These files may be created or modified by use of the DESIGN command. This is a faster but less powerful (fewer subcommands) version of the CTSS input-edit command, and it treats all information as a string of characters without regard to its ultimate usage. Several other general housekeeping commands are available. These are: LISTF, which allows the user to list all or a subset of his file directory; PRINTF, which prints out the contents of a file; and ERASE,which deletes one or more of the users files. The command structure of CIRCAL-II is shown in Fig. 2.

A data structure for CIRCAL-II has been designed and implemented. This structure is sufficiently general to allow analysis by any number of separate routines. This work is the result of a Master's Thesis by James R. Stinger and is discussed in Section C.

CIRCAL-II may employ any one of several analysis subprograms which are stored in CIRCAL'S files. The first such program, a basic frequency analysis routine, was written in order to fully determine the

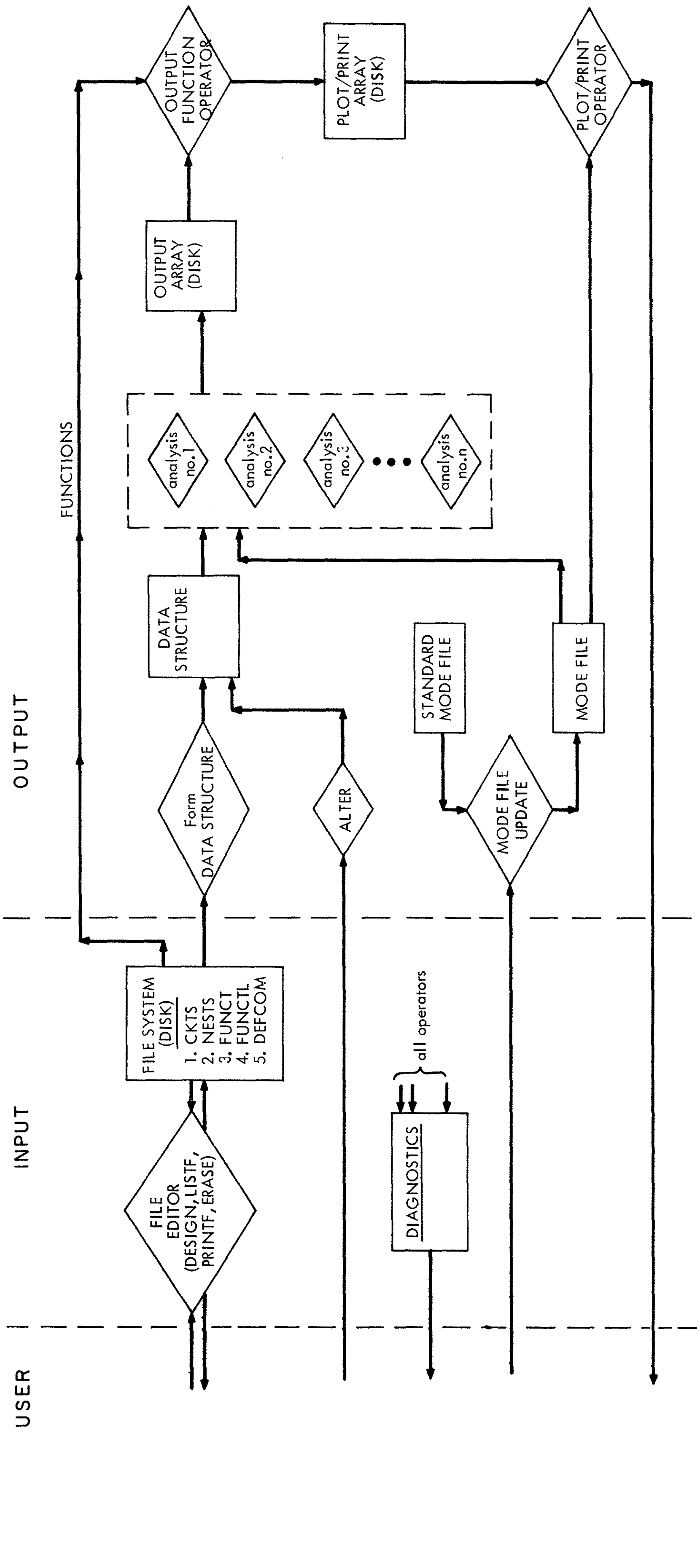


Fig. 1 Basic Structure of CIRCAL-II

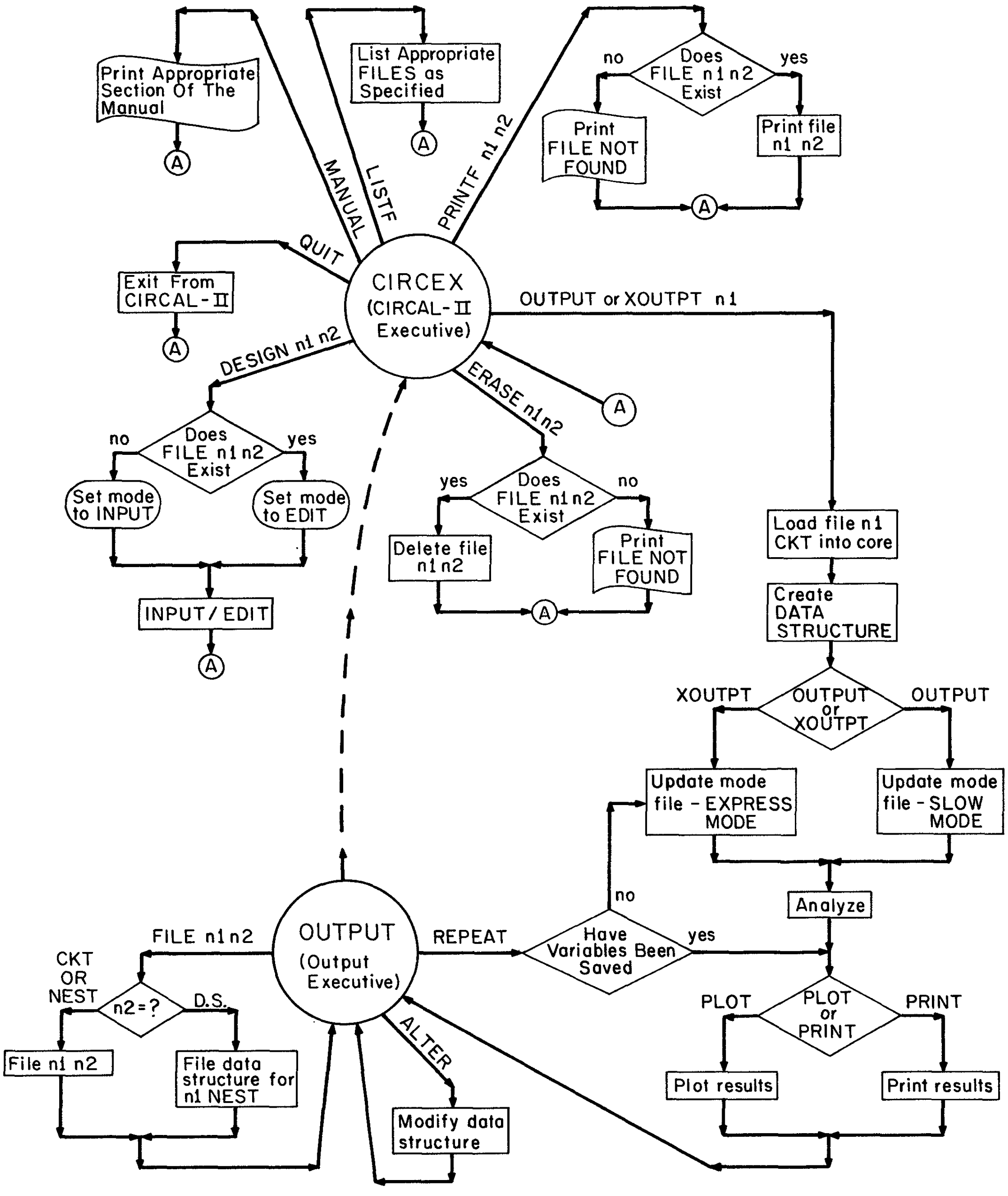


Fig. 2 The CIRCAL-II Command Structure

generality of the data structure and the overall modularity of the system. Observe that only three man-weeks were required to develop this program, as opposed to several man years which were spent on CIRCAL-I. This confirms one of the basic objectives for the design of CIRCAL-II, i.e., the realization of savings by a standard implementation of the large number of common "overhead" programs present in on-line circuit design. Networks of up to 100 elements and 30 nodes, with several levels of nesting, have been already analyzed, using the above analysis program. The limitation of network size arises from the amount of space in core available over and above that required by the programs. Although this limitation will not be present in MULTICS, a scheme is presently being implemented to dynamically load the necessary programs so as to increase the amount of available storage at any stage of operation.

The above analysis program forms the capacitance, conductance, and reluctance matrices of the network from the CIRCAL-II data structure. At each frequency, the resultant admittance matrix is then solved by Gaussian Reduction, and back substitution to determine nodal voltages. Those voltages (real and imaginary) desired for output are then entered into the standard output array of CIRCAL-II. This array of output variables is, in turn, saved on disk, so that the amount of available space in core places no limit on the number of analysis points which can be calculated. At present, a PLOT/PRINT routine is available, which operates directly on this array to present the data to the user. Future versions of CIRCAL-II will incorporate an intermediate operator which will be able to calculate and display any function (e.g. logarithm) of the computed output variables.

For purposes of designing a circuit, several options are available to the user. He may simply change a parameter and REPEAT the process; or he may give the ALTER command in order to change the topology of the network (on the CIRCAL-II data structure) prior to re-analysis; or finally he may return to the input phase and begin work on a new circuit.

Future work will entail the following: First, a number of different analysis routines will be implemented to broaden the scope of CIRCAL-II. These will include linear-time-domain, nonlinear-transient and statistical-sensitivity analysis. Second, CIRCAL-II will be used as a forum for implementing two techniques which are nearing completion of their theoretical stage (see Sections 3 and 4 below). These techniques involve the solution of nonlinear networks by tearing and by a new recursive-structure method of analysis. Finally, an area of interest centering on definitional commands will be implemented. These (DEFCOM) features will act as a pseudo-user and will allow operations such as network optimization to be done automatically.

C. THE CIRCAL-II DATA STRUCTURE

Professor M. L. Dertouzos

Mr. J. R. Stinger
Research Assistant

This section describes the development of the CIRCAL-II "data structure former" which was discussed in the preceding section.

Data structures produced by the data-structure former consist of four basic primitives:

1. The network bead (contiguous block of memory registers)
2. The element bead
3. The node bead
4. The parameter bead

For each element in the circuit there is a corresponding element bead, and for each node there is a corresponding node bead which contains information pertaining to that element or node. If during design of a network the value of one or more elements in the network is left unspecified, i. e., given a variable name instead of a value, then the data structure will contain a parameter bead for each such distinct variable. These beads will, in turn, contain the desired values of these variables, when they are specified, prior to analysis of the network.

To make the circuit information readily available for analysis, the above element and node beads are connected together by pointers (addresses of words in memory) to form rings. The members of each of these rings share some common property, for example, all resistors are placed in one ring, and all external nodes are placed in another, so that by progressing through a ring, all elements or nodes having that property will be encountered and read by a subsequent analysis subprogram. The current implementation of the data structure has a total of 15 rings. To make each of these rings easily accessible, each circuit has a corresponding network bead which contains pointers to the entry point of each ring. Thus, the general form of the data structure is as shown in Fig. 3.

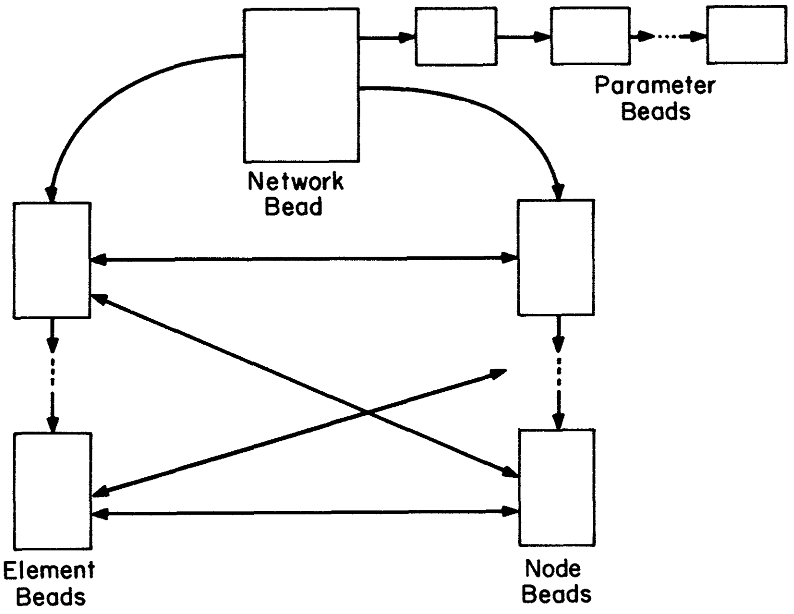


Fig. 3 General Form of the CIRCAL-II Data Structure

Since the user is allowed to create nests (circuits with external nodes treated as single elements), this data structure is recursive. That is, for each nest in the circuit a data structure of the form shown above is created and the element bead for each instance (occurrence in a circuit) of this nest contains a pointer to this "generic" data structure. No restriction is placed on the depth of nesting allowed in a circuit other than that imposed by core memory size. Observe that although there may be many instances of a nest in a circuit, there will only be one generic data structure for that nest, thus resulting in efficient use of the space available for the network's data structures.

Observe further that the form of the data structure was made as general as possible to enable coupling to many different types of analysis routines. Not all of the analysis routines envisioned for inclusion in CIRCAL-II, however, will need all of the features (i.e., rings) of the data structure to perform analysis of the circuit. Therefore, to each analysis routine there corresponds an analysis mask which indicates those features of the data structure needed by that analysis routine. This mask is used in creating the data structure, making the creation of the data structure a more efficient process, as well as making the data structure variable with respect to the analysis routine used.

Finally, provisions have been made in implementing the data structure former of CIRCAL-II, to allow the writer of an analysis routine to add new rings to the data structure should the existing rings be inadequate.

D. A RECURSIVE APPROACH FOR THE COMPUTER ANALYSIS OF NONLINEAR NETWORKS

Professor M. L. Dertouzos
Mr. H. L. Graham
Research Assistant

1. The Inversion Problem

Amnesic systems, such as nonlinear resistive networks and memoryless block-diagram systems, are of wide interest to general system analysis and design. The solution of these amnesic systems constitutes the general functional inversion problem. More explicitly, a network or system is easily characterized by a direct function, $\underline{f}$, which maps the response vector, $\underline{y}$, into the system excitation vector, $\underline{u}$, (i.e., $\underline{u} = \underline{f}(\underline{y})$). The functional inversion problem consists of determining the inverse function, $\underline{f}^{-1}$, which maps the excitation into the response, (i.e., $\underline{y} = \underline{f}^{-1}(\underline{u})$).

At present, in the analysis of nonlinear networks and systems, the direct function is not inverted at all points of its domain. Instead, each time a different excitation is encountered the inverse function is evaluated, for that vector, by some iterative, trial-and-error procedure which involves a sequence of direct-function evaluations.

The object of this research is the development of methods for the inversion of direct functions in a direct non-iterative manner. With such methods, an initial overhead would be expended for construction of the inverse function. Subsequently, the response to any excitation vector could be easily evaluated. In this research, primary emphasis is given to functions that characterize classes of nonlinear resistive networks.

Toward this objective an algebra of functions was developed by which a network's inverse function can be expressed in terms of the basic network-element functions,* and general algebraic techniques were developed for obtaining the inverse function from the network equations. Function representations obtained in this manner generally contain operations on functions of n-1 variables where n is the number of network nodes. Since the computational effort expended depends directly on the order (number of variables) of the functions being manipulated, practical considerations dictate the need for representation of the lowest possible order. For this reason, the concept of network order was developed and investigated.

2. Network Order

Network order is based on an expansion of the familiar class of series-parallel networks. Such networks have an elementary recursive structure which provides a direct means of constructing the inverse function. This basic class of networks is defined as follows:

A network is series-parallel if and only if:

a. It is a two-terminal element, or

b. It can be obtained by the series or parallel connection of two series-parallel networks.

The "construction rules" shown in Fig. 4 correspond to this definition.

*"Computer-Auded Electronic Circuit Design" (Part I), Electronic Systems Laboratory Status Report ESL-SR-322, September, 1967.

Note from the set of constructions that a series-parallel network is always defined with respect to two terminals, designated as

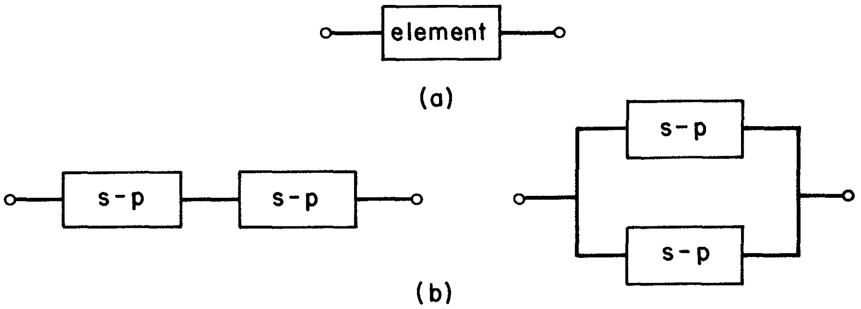


Fig. 4 Series-Parallel Construction Rules

the external terminals. If a network is series-parallel, the function relating network variables to a driving voltage (or current) at the external terminals can always be represented in terms of first order functions.

This structural concept of networks can be extended to higher dimensions as follows: Denote the class of series-parallel networks as N_1. The general class of networks, N_k, based on the formation of k-terminal networks from pairs of k-terminal networks is defined as follows:

A network is in class N_k if and only if:

a. It is in class N_{k-1}, or

b. It can be constructed by connecting two networks of class N_k and has, at most $k + 1$ external terminals.

For example, the construction rules for networks of the next higher order (order 2) are illustrated in Fig. 5.

The inverse function of any member of N_k can be represented in terms of functions of order k or less. Thus the order of a network n is defined to be ℓ if $n \in N_\ell$ and $n \notin N_{\ell-1}$. (Note that N_1 N_2 N_3 ... etc.)

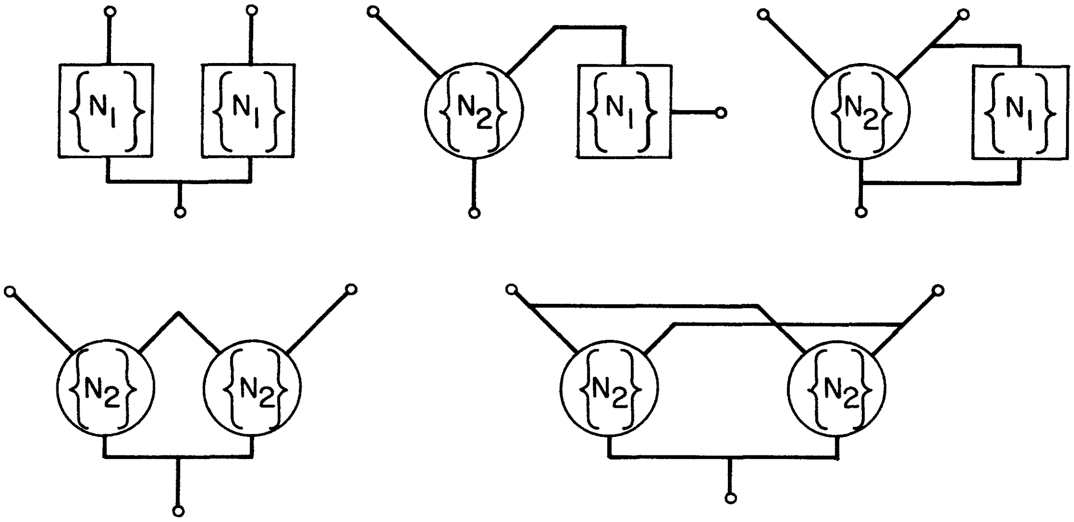


Fig. 5 Order-2 Construction Rules

Practical use of the order concept requires a procedure for determining the order of a network. An exhaustive, though impractical, procedure does exist for any finite network since all possible constructions can be enumerated. A more feasible procedure has been developed which determines if a network of n-nodes and b-branches is of order 2 in, at most, a time proportional to $b^2 \binom{N}{2}$. Improvements on this procedure are the objectives of present and continued research in this area.

Expressions for the order of specific types of networks have been determined. A result of particular importance shows that the order of a complete graph on n-nodes is the integer q such that $2/3\ (n-1) \leq q < (2/3\ n)$. This, in turn, provides an upper bound for the order of any network on n-nodes.

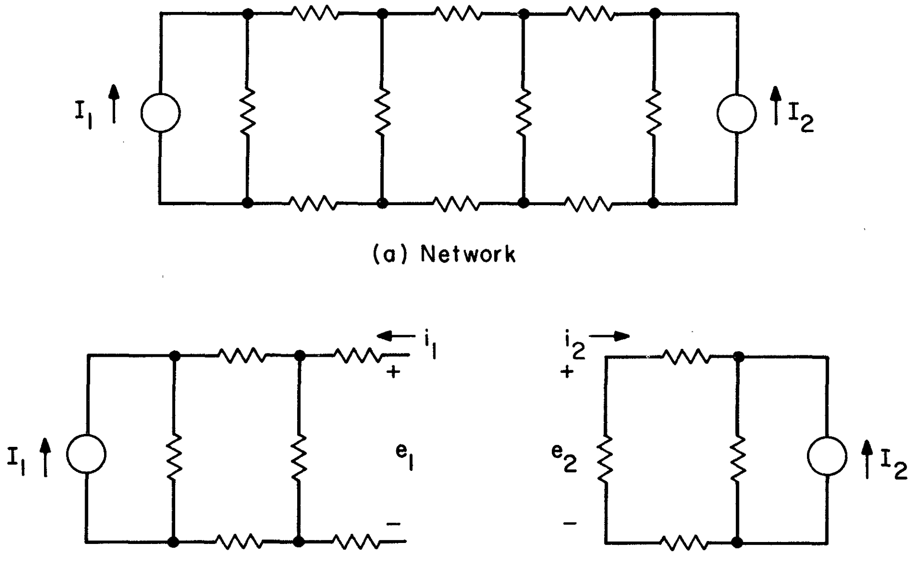
Future research is primarily concerned with the improvement of order-determining techniques, with the implementation of these theoretical concepts into practical analysis procedures, and with the coupling of these procedures to the CIRCAL-II program.

E. TEARING TECHNIQUES

Professor M. L. Dertouzos
Mr. C. W. Therrien
Research Assistant

Tearing is a method for solving networks by splitting them in pieces, solving each of the pieces, and then constructing the solution of the entire network from the solutions on the pieces. A question important to the design of on-line circuit analysis programs is, "How should networks be torn in order that they may be solved with a relatively small computational effort?" A formal theory of tearing including a tearing model and related algorithms is under development to treat this general question.

Tearing of a network is accomplished by separating the network "at some nodes" into a number p of disjoint pieces (see Fig. 6). Such a separation is termed a "p-tear" of the network. In the case of a 2-tear, the nodes effecting the separation are called a "separation set". The solution of the network is carried out in two stages. First, each of the pieces is solved, considering the currents and voltages at the nodes of tearing as external variables. Next, the pieces are



(a) Network

(b) a 2-tear

Fig. 6 2-Tear of a Network

reconnected and the resulting currents and voltages at the nodes of tearing are eliminated from the equations representing the solutions of the pieces.

The tearing model reflects the two stages of the method as follows. The model postulates the existence of certain "computation functions" $f(x)$ with properties described in Reference 1 that provide a measure of the computation involved in solving a network. The argument x may represent any one of several possible measures of the size of the network graph (see Ref. 1), but will be taken here to be the number of nodes in the network. Examples of computation functions are the number of computer operations required to invert the nodal admittance matrix of a linear network or the number of iterations required to compute the solution of a nonlinear network by a relaxation algorithm. The computation required to solve an N-node network by the method of tearing can then be expressed in terms of the computation functions. The ratio of this computation to $f(N)$, the computation required to solve the network without tearing, is called the computation ratio C and for a 2-tear is given by

$$C = \frac{f(m) + f(N - m + k) + f_N(k)}{f(N)}$$

and defined over a region $2 \leq k \leq \min [m, N - m]$. Here m is the number of nodes in one section and k is the number of nodes in the separation set. The first two terms in the numerator represent the computation for solving each of the pieces while the last term represents the computation for the interconnection problem. The subscript N denotes that this computation function is additionally dependent on the total number of nodes in the network. Both f and f_N have the general properties of computation functions given in Reference 1.

Investigation of the computation ratio C results in several general properties of 2-tears of N-node networks. For example, it can be shown that a lower bound on the reduction of computation achievable by tearing is

$$C\Big|_{\min} = \frac{2f(N/2 + 1) + f_N(2)}{f(N)}$$

Similar results can be obtained for the general case of a p-tear, (see Ref. 1).

Unfortunately, the computation ratio says nothing about the best way to tear a particular network, since the tears indicated as optimum may not exist in a particular network undergoing analysis. One technique for finding the optimum 2-tear of a network is to plot the locus of all possible tears of the network in the above k-m space, and to minimize C over this region. This technique is demonstrated in Fig. 7 for an n-by-n-node square network graph. A disadvantage of this method is that it is generally difficult to find the locus of tears by other than exhaustive means.

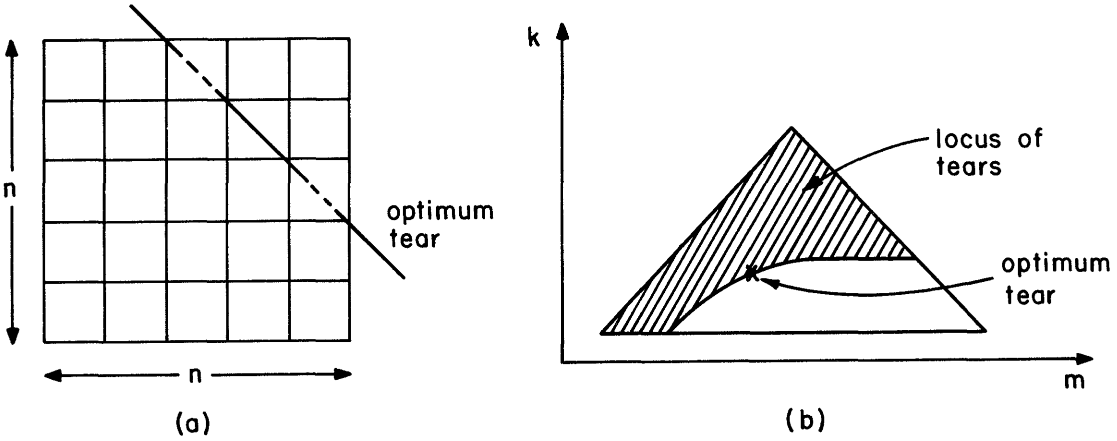


Fig. 7 Locus of Tears for a Grid Network

Several other algorithms for locating 2-tears, based on transport networks and statistical techniques, have been tried with reasonable results. Currently, an algorithm for finding two sets of maximally unconnected nodes is being considered.* The algorithm seems to be slower than some of the ones already developed, but it has the advantage that there are no degenerate networks for which it fails to find a good 2-tear (unless the network has none). Present work is directed toward speeding up this algorithm and extending its application.

*Actually one searches for two maximally connected sets in the complement graph of the network.

F. LOTUS: ON-LINE SIMULATION OF BLOCK-DIAGRAM SYSTEMS

Professor M. L. Dertouzos
Mr. M. E. Kaliski
Fellowship Student
Mr. K. E. Polzen
Research Assistant

This work involves the development of a general approach for on-line simulation of a variety of block-diagram systems. These systems may be analog, digital, or hybrid; with memory or memoryless; explicit or implicit (i.e., possessing loops of elements); vector or scalar; or of a more specialized nature, such as dynamic systems with integrators as memory elements; and simulation may be conducted in one or more dimensions, such as time and/or space. An on-line simulation program, LOTUS [1], implemented through the facilities of Project MAC, has been developed to realize the syntactical and organizational aspects of simulating this wide class of systems.

The following objectives were pursued in this development:

1. provide a natural and convenient man-machine interface, which would allow a user, with limited programming knowledge, to simulate and analyze his system in an on-line environment, and

2. establish a basic set of fundamentals for system representation and analysis, which encompass a broad class of applications.

In terms of the second objective, an essential feature of LOTUS is the creation of data structures which characterize the block-diagram system and which are general enough to allow efficient evaluation of the class of systems described above. These systems may exhibit a recursive interconnection structure in the sense that subsystems may be "nested" into single elements and connected to other subsystems to form a larger system, as is the case in CIRCAL-II. System evaluation is therefore conducted by a recursive algorithm which traces through the data structures and sub-data structures, until a set of primitive operators (i.e., operators composed of no further subsystems) have been evaluated.

The present version of LOTUS simulates explicit systems with inputs and outputs defined on the real numbers. The user has the ability to define these systems through an appropriate sequence of commands which interconnect primitive elements and/or nested structures into any configuration in which no two outputs are connected. Through other on-line commands which are similar to those of CIRCAL-II, he may edit or modify the structural form of the system he is analyzing. The program has been written modularly to permit future growth.

Work currently in progress is in two areas:

1. Making the current, limited, version of LOTUS more efficient.
2. Using the fundamentals of system representation to extend the power of the program to handle all systems in the class specified above.

G. PUBLICATIONS OF THE PROJECT

1. Current Publications

a. Reports

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-337, January, 1968.

b. Thesis Proposal

Therrien, C. W., Tearing of Networks, proposal for Ph. D. dissertation, Department of Electrical Engineering, M. I. T., March, 1968.

c. Theses

Kaliski, M. E. and Polzen, K. P., "LOTUS, On-Line Simulation of Block Diagram Systems," Master of Science joint thesis, Department of Electrical Engineering, M. I. T., January, 1968.

Stinger, J. R., "A General Data Structure for On-Line Circuit Design," Master of Science thesis, Department of Electrical Engineering, M. I. T., January, 1968.

2. Past Publications

a. Reports

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-225, December, 1964.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-245, June, 1965.

Dertouzos, M. L. and Therrien, C. W., "CIRCAL: On-Line Analysis of Electronic Networks," Report ESL-R-248, December, 1965.

Dertouzos, M. L. and Santos, P. J., Jr., "CADD: On-Line Synthesis of Logic Circuits," Report ESL-R-253, December, 1965.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-256, December, 1965.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-274, June, 1966.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-298, January, 1967.

"Computer-Aided Electronic Circuit Design," Part I, Status Report ESL-SR-322, September, 1967.

b. Technical Papers and Conference Participation

Reintjes, J. R. and Dertouzos, M. L., "Computer-Aided Design of Electronic Circuits," WINCON Conference, February, 1966, Los Angeles, California

Reintjes, J. F., "The Role of Computers in Modern Design Technology," Conference on Computer-Aided Design, University of Wisconsin, May 3-4, 1966.

Dertouzos, M. L. and Graham, H. L., "A Parametric Graphical Display Technique for On-Line Use," presented at Fall Joint Computer Conference on November 8, 1966. Published in the Conference Proceedings of the FJCC.

Therrien, C. W. and Dertouzos, M. L., "CIRCAL: On-Line Design of Electronic Circuits," presented at the NEREM Show and published in NEREM record November 9, 1966.

Notes : (1) Professor Dertouzos was Chairman of the Computer-Aided Electronic Circuit Design Session at the NEREM 1966 Conference, Boston, Massachusetts.

(2) CIRCAL-I was used from Munich, Germany via TELEX, June, 1966, in connection with a series of ten lectures by Professor Dertouzos at Siemens, Halske.

Katzenelson, J., "AEDNET : A Simulator for Nonlinear Networks," Proceedings of the IEEE, Vol. 54, No. 11, November, 1966, pp. 1536-1552.

Therrien, C. W., presentation on CIRCAL at N. Y. U. Conference on "Network Analysis by Computer Symposium," New York University, January, 1967.

Dertouzos, M. L., Panelist in panel discussion, "On-Line Versus Batch," held at the NASA Computer-Aided Circuit Design Seminar, April 11-12, 1967, Cambridge, Mass.

Dertouzos, M. L., "PHASEPLOT : An On-Line Graphical Display Technique, IEEE Transactions on Electronic Computers, Vol. EC-16, No. 2, April 1967, pp. 203-209.

Dertouzos, M. L. and Fluhr, Z. C., "Minimization and Convexity in Threshold Logic," Seventh Annual Symposium on Switching Circuit Theory and Logical Design, Berkeley, California, 1966; IEEE Transactions on Electronic Computers, Vol. EC-16, No. 2, April 1967, pp. 212-215.

Dertouzos, M. L., "CIRCAL: On-Line Circuit Design," Proceedings of the IEEE, Vol. 55, No. 5, May, 1967, pp. 637-654.

Dertouzos, M. L. and Graham, H. L., "A Parametric Graphical Display Technique for On-Line Use," M. I. T. Project MAC Seminar, May 19, 1966.

Evans, D. S. and Katzenelson, J., "Data Structure and Man-Machine Communications Problems," Proceedings of the IEEE, Vol. 55, No. 7, July, 1967, pp. 1135-1144.

Dertouzos, M. L. "An Introduction to On-Line Circuit Design," Proceedings of the IEEE, Vol. 55, No. 11, November, 1967, pp. 1961-1971; also Proceedings of the Fifth Allerton Conference on Circuit and System Theory, October 4-6, 1967 (Invited Paper).

Dertouzos, M. L., "Panel Discussion on Computer-Aided Design," Proceedings of the IEEE, Vol. 55, No. 11, November, 1967, pp. 1777-1778.

Dertouzos, M. L., Chairman of Session "Computer-Aided Circuit Design; A Critical Appraisal," NEREM 1967, Boston, November 3, 1967.

Dertouzos, M. L., Co-chairman of and Lecturer at Industrial Liaison Symposium on "Computer-Aided Circuit Design," M.I.T., October 3, 1967.

Dertouzos, M. L., Co-chairman of and Lecturer at M.I.T. Summer Course 6.56S on "On-Line Circuit Design," M.I.T. July 6-July 13, 1967. (One set of Proceedings issued at that course.)

Dertouzos, M. L., Talks and Demonstration on On-Line System and Circuit Simulation to:

a. EE Department, Catholic University of America, Washingston, D.C., November 10, 1967.

b. IEEE Computer Group, Boston, October 11, 1967.

c. Theses

Dvorak, A. A., "An Input-Output Program for Electronic Circuits Using a CRT," Bachelor of Science Thesis, Department of Electrical Engineering, June, 1965.

Santos, P., "CADD, A Computer-Aided Digital Design Program," Master of Science Thesis, Department of Electrical Engineering, June, 1965.

Therrien, C. W., "Digital-Computer Simulation for Electrical Networks," Master of Science Thesis, Department of Electrical Engineering, June, 1965.

Fluhr, Z. C., "Single-Threshold Element Realizability by Minimization," Master of Science Thesis, Department of Electrical Engineering, August, 1965.

Olansky, K. J., "A Low-Cost Teletype-Operated Graphical Display," Master of Science Thesis, Department of Electrical Engineering, August, 1965.

Gertz, J. L., "A Graphical Input-Output Program for Digital System Simulation," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Graham, H. L., "A Hybrid Graphical Display Technique," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Meltzer, J. R., "CIRCAL: An Input for Nonlinear Elements," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Taubman, C. N., "Computer Analysis of Electrical Analog Distribution Systems," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Walpert, S. A., "An Output Program for Representing Electrical Signals," Master of Science Thesis, Department of Electrical Engineering, June, 1966.

Edelberg, M., "A Dual Approach to Threshold Decomposition of Boolean Functions," Master of Science Thesis, Department of Electrical Engineering, June, 1967.

Willems, J. D., "Synthesis of Logical Functions with Restricted Threshold Elements," Electrical Engineer Thesis, Department of Electrical Engineering, June, 1967.

Smith, T., "Nesting of Networks for Computer-Aided Circuit Design," Bachelor of Science Thesis, Department of Electrical Engineering, June, 1967.

d. Motion Picture

CIRCAL: Computer-Aided Electronic Circuit Design, January, 1966.