

AN ON-LINE CIRCUIT DESIGN SYSTEM

NAN-HUNG KUO, CHYAN GOEI CHUNG and YEN SON HUANG

College of Engineering, National Chiao Tung University

(Received 6 December 1973)

Abstract—*The purpose of this paper is two folds:*

- (1) *To introduce an on-line computer system for circuit analysis and design, which will be set up in Chiao-Tung University;*
- (2) *To outline the basic method employed by us for developing an entire software system from system design to mask preparation.*

1. INTRODUCTION

Due to great computational speed and simulation ability, digital computers are being used in the difficult and tedious job of designing and analyzing new systems¹⁷. The three main areas of design automation of new systems are preconstruction analysis, hardware design, and implementation. If a new computer system is to be designed, the area of software generation should be added. For example, Fig.1 is the design procedure of a new electronic circuit system. Preconstruction analysis deals mainly with the use of system simulators in trying to determine an optimal functional configuration for a new system, and then decomposes the system as function boxes. Problems included in the hardware design are circuit analysis and design, new component design, and generation of documentation. And physical implementation problems are the assignment of components to circuit card, the assignment of circuit cards to blackboards, the placement of circuit cards on blackboards, and the interconnection of terminals.

The advantages of design automation are as follows¹⁷:

- 1) a possible reduction in costs,
- 2) increased problem analysis capabilities,
- 3) an increase in man creative ability through man-machine interaction,
- 4) increased speed of design,
- 5) change control and documentation,
- 6) an increase in the efficiency of the available manpower.

The last two items are probably the main factors which influence what design procedure should be automated first.

These steps can relieve the design engineers of many tedious and errorprone chores and allow them to do more creative design.

2. OUTLINE OF DESIGN PROCEDURE

Fig.1 shows three different aspects of circuit design which will be carried out in Chiao-Tung University within three years:

- (1) System design in terms of functional boxes (preconstruction analysis); (2) Box

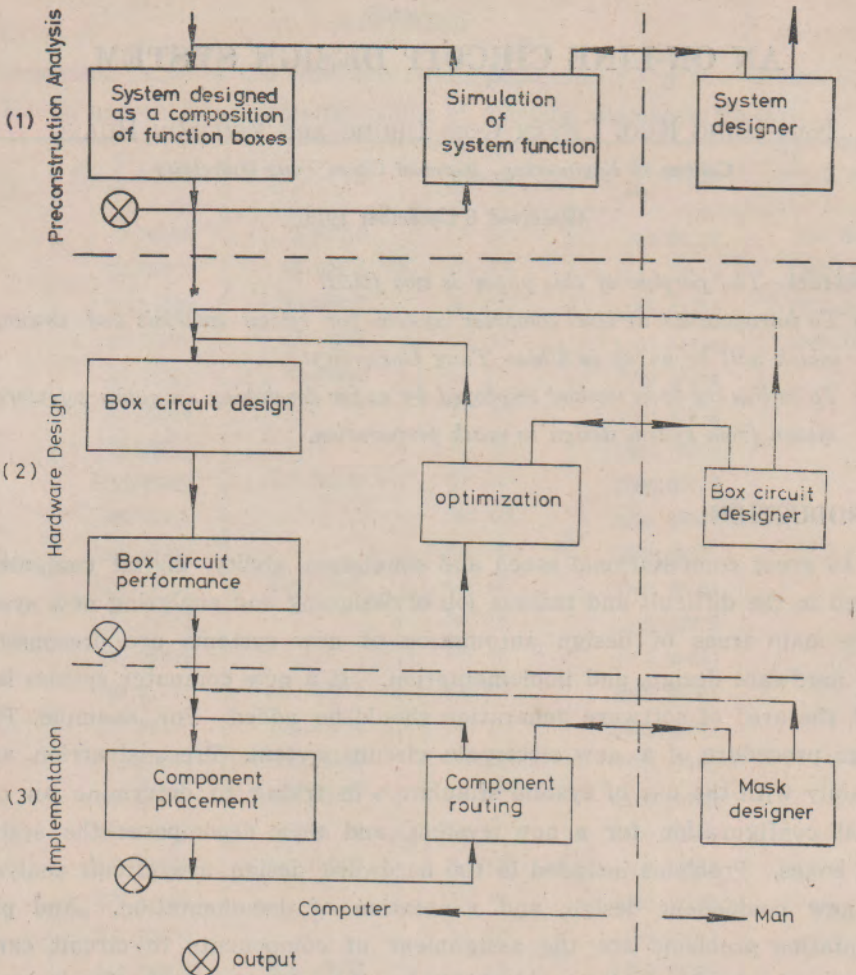


Fig. 1. The Procedure of A On-Line Circuit Design.

circuit design by adjusting network topology and parameters (hardware design); and (3) Mask design for integrated circuits by suitably component locating and routing.

The highest level of design uses a general purpose systems simulator to evaluate the system characteristics, and then decomposes the system in terms of ideal function boxes^{9,18,19}. This decomposition may be obtained intuitively or automatically. In this aspect, the computer plays merely as a simulator or error checker.

In the box circuit design, the network under study is analyzed so that its performance is evaluated for review by the designer who changes the parameter vector or even network topology to meet the desired network function. The process of adjusting parameter vector efficiently is called "Optimization". Both analysis and optimization are carried out by the computer, with occasional human decisions on optimization strategy¹.

Complete design automation of masks is far from reality. The successful design still relies heavily on occasional designer's intervention on decisions regarding component placement and subsequent interconnection routing¹⁰.

In conventional processing, the designer may wait for several days to readjust the trial network for further improvement. With graphical input-output, the designer's decision on modification of the network may be immediately reached by inspection upon analysis results. A computer system with such a man-machine interaction will be introduced in the next section.

3. THE ON-LINE UTILITY

The on-line utility consists of computer hardware and software. The hardware configuration is demonstrated in Fig. 2, which shows the central computer DEC/10 interfaced with a satellite computer MODCOMP. Through the teletypewriter, the designer can communicate with the satellite computer which directs all problem-defining activities at the CRT display. The satellite computer composes and modifies circuit, and converts informational structure. When the circuit is ready for numerical analysis, the central computer is called into services, and analysis results are returned to the satellite computer for display. In this configuration most demanding analysis tasks are performed in the central computer.

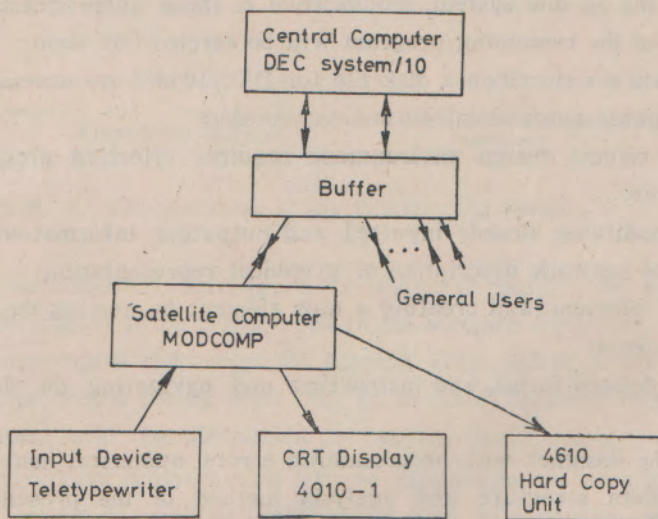


Fig. 2. The Hardware Configuration of The On-Line System.

The software system shown in Fig. 3 consists of three parts:

Parts A: A input-output cotrolling program written for MODCOMP. The program is responsible for both network specification and result displays;

Part B: A supervisor program written for DEC/10 to do data conversion, command execution, and output direction;

Part C: The main program written for DEC/10 to perform circuit analysis, component design, etc.

The input/output controlling program will be written in a modified graphical input-output language to be developed at Chiao-Tung Universitp. The supervisor program will be written in the assembly language of DEC/10. Regarding the main program, a part has been written in FORTRAN IV mixed with a few assembly language for batch

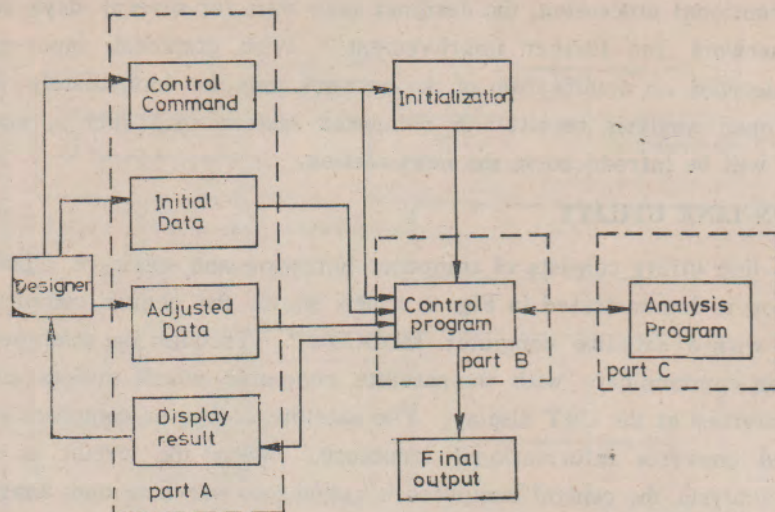


Fig. 3. The Software Structure of The On-Line System.

processing. For the on-line system, modification of these subprograms are required and completion of the remaining program will be carried out soon. The supervisor and main program are stored on a disk file for DEC/10 and are accessed and used by the central computer automatically whenever needed.

The on-line circuit design environment requires effective program to do the following services:

- (1) Inputting, modifying already inputted, and outputting information: This information may be network description or graphical representation;
- (2) Define new element, and creating a new element by nesting the already defined elements together;
- (3) Identifying defined terms, and instructing and navigating the designer for the system;
- (4) Supplying the designer with programming errors, numerical instabilities, etc.

Thus, both data structure and analysis method of the present programming system must be designed for satisfying the above requirements.

4. OPERATING SYSTEM AND PROGRAMMING STRUCTURE

In order to make the man-machine interaction effective and the on-line circuit design program powerful, an interrupt system and parallel processing is adopted. Normally, the satellite and central computers execute their own tasks parallelly. When user types in a command from teletypewriter, the satellite computer is interrupted to accept it and interpret the command, and then take action. If the satellite computer requires to order the central computer to do analysis, to output certain information, to do some disk operation, to cancel current executed task or to execute other task etc., then the satellite computer should issue a command to the central computer and interrupt it. After finishing the command processing, the central computer will continue its normal operation or change to execute another

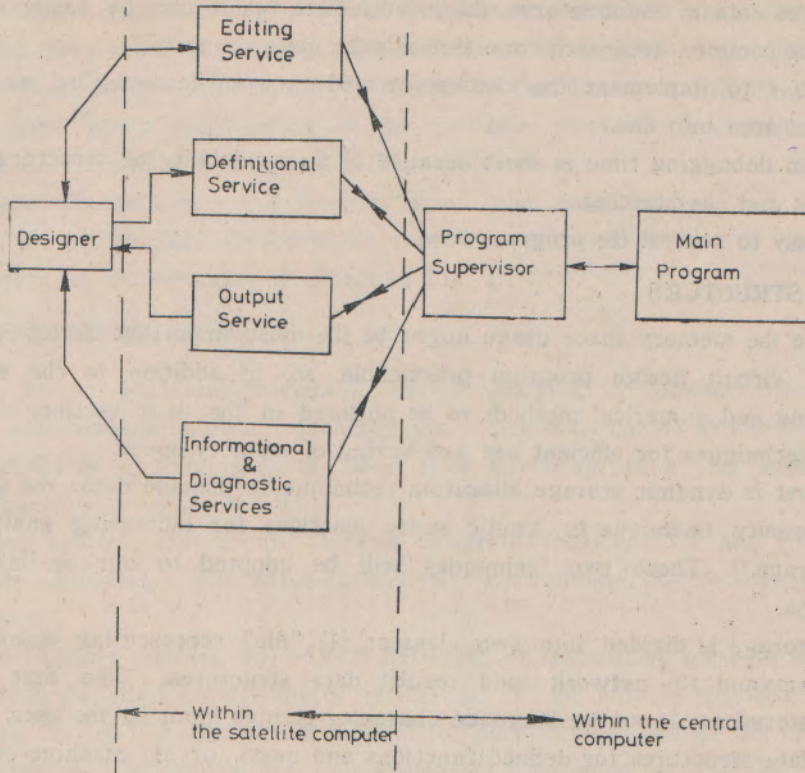


Fig. 4. The Flowchart of The Programming System.

task according to the command activity. Conversely, if the central computer has some output or information to be transferred to the satellite computer, the satellite computer will be interrupted and accept the transfer data. After accepting the data, the satellite computer should decide what task to be done consecutively. This kind of interactive system will be developed in our on-line system.

Moreover, the structure programming will be adopted in the satellite computer as well as central computer to minimize developing time, to maximize memory space usage and program flexibility, and to make program control easier.

The main features of structure programming are as follows:

- (1) Main program is short and contains a set of simple statements and subroutine calls;
- (2) Each subroutine is a self-contained logical module;
- (3) Each subroutine execution control should return to the main program;
- (4) Avoiding as possible as subroutine-call-subroutine and GO TO statements, particularly jump-upward GO TO statements.
- (5) The linkage between the main program and the subroutines is through common area.

Such structure programming as described above will give the following advantages:

- (1) Program overlay can be used effectively;
- (2) On line capability is increased a great deal. Because every program uses or

calculates data in common area, the intermediate result can be easily obtained from the common area each time through the main program.

- (3) It is easy to implement the checkpoint and recover features by storing the common area into disk.
- (4) Program debugging time is short because of the readability of structure programming and easy testing.
- (5) It is easy to control the program flow.

5. DATA STRUCTURE

Because the memory space usage might be the most important factor to make the on-line circuit design program practicable, so, in addition to the structure programming and numerical methods to be outlined in the next section, there are two other techniques for efficient use and saving of memory space.

This first is dynamic storage allocation technique to manage data: the second is network sparsity technique to handle spare matrices for increasing analysis and saving storage.²² These two techniques will be adopted to our on-line design program too.

Data storage is divided into two classes: (1) "file" representing standard and defined items, and (2) network (and result) data structures. The first class is generally stored as (1) source-language character strings input by the user, (2) intermediate data-structures for defined functions and nests, or (3) machine-code data for standard elements and standard functions.

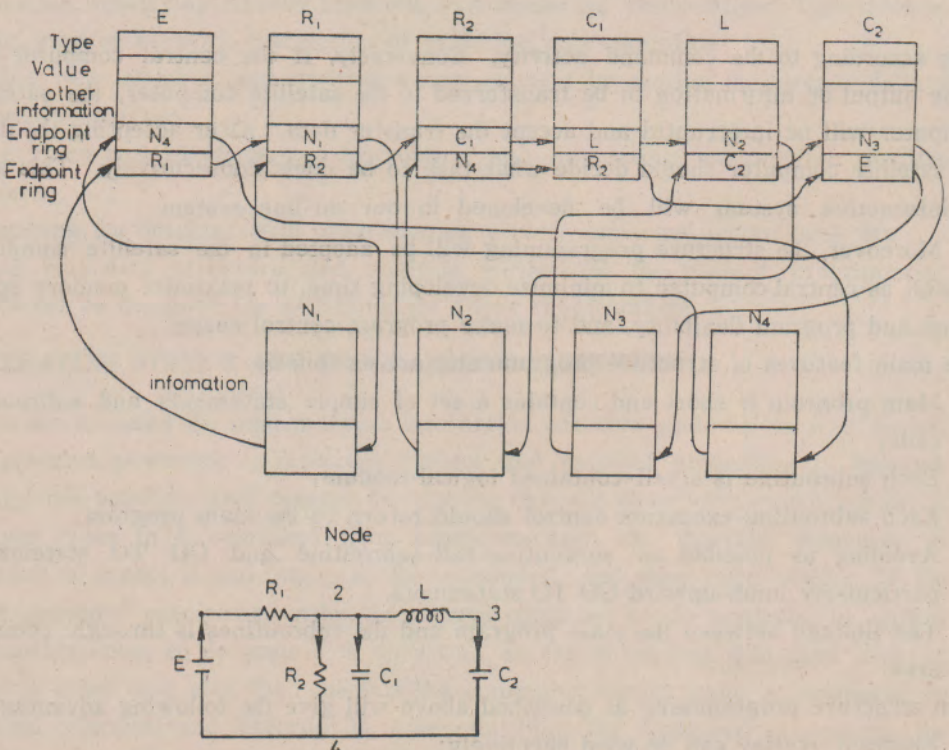


Fig. [5. Network Diagram and Corresponding Data Structure.

The second class is stored in blocks called the trailer which has arbitrary length. There are three kinds of trailers: node trailers, component trailers, and non-circuit trailers. Most circuit information is stored in the first two trailers. Noncircuit trailers give other specification of the problem such as frequencies desired, computation requestes, and output option selected.

Because of effective using momory space and easy deleting and adding new element, we wliil use list structure to represent network structure. An example to describe the list structure is shown in Fig. 5.

6. NUMERICAL METHODS

The entire program consists of circuit analysis, component placement and routing, and component design subprograms. Each subprogram will be written under the principles of minimizing development time, execution time and memory space, and maximizing program flexibility.

In our system, the circuit analysis program includes DC, AC, transient and sensitivity analyses. For efficient analysis we combine the following advanced techniques:

(A) state variable method, (B) adjoint network for calculating network sensitivities, (C) network partition for parameter variations, (D) the varable step-size integration technique, (E) the Fletcher-powell method for optimization, and so on. Most of the above-mentioned techniques are well known and so need no further oxplanation. we shall introduce only the recently developed techniques in the succeeding paragraphs.

The first step for analyzing a nonlinear circuit by a digital computer is to find a proper tree and then to formulate the state equations in its normal form

$$\dot{x} = f(x, u, t) \tag{1}$$

$$y = g(x, u, t) \tag{2}$$

where x , u and y denote state, input and output vectors. If Eq. (1) is stiff and if an ordinary numerical technique is applied for time-domain solution, then excessively small time steps have to be used to obtain a meaningful appoximate solution since otherwise numerical instability results. Recently it has been shown²³ that the second-order implicit integration formula, are stiffly stable:

$$\begin{aligned} X_{n+1} = & -X_{n-1} \frac{h_1^2}{h^2(2h_1+h_2)} + X_n \frac{(h_1+h_2)^2}{h_2(2h_1+h_2)} \\ & + X_{n+1} \frac{h_1(h_1+h_2)}{2h_1+h_2} \end{aligned} \tag{3}$$

where $X_n = X(t)|_{t=t_n}$, $h_1 = t_{n+1} - t_n$ and $h_2 = t_n - t_{n-1}$. Besides stiffly stable, the above scheme for integration has the following features: the variable step size and the second order of accuracy. The local truncation error of eq. (3) is

$$E_{n+1}^* = \frac{h_1^2(h_1+h_2)}{6(2h_1+h_2)} X^{(3)}(t^*) \tag{4}$$

where $t^* \in [t_n, t_{n+1}]$. Because the magnitudes of the components of X can range over several orders of magnitude in the interval $[t_o, t_f]$ where t_o is the initial time and t_f is the final time. To maintain the same relative accuracy for all the components of the approximate solution of (3), each component of E^*_{n+1} is divided by the corresponding component of X_{n+1} . The resultant vector is denoted by \underline{E}^{**}_{n+1} . Let l^{**} is the largest magnitude of \underline{E}^{**}_{n+1} . If $l^{**} > K_1$, the step size is halved and X_{n+1} is recomputed. If $l^{**} < K_2$, the step size is doubled. If $K_1 \geq l^{**} \geq K_2$, the step size is retained. The K_1 and K_2 are chosen to satisfy the ratio $K_1/K_2=8$, K_1 is selected by the program error.

The best approach at present to find the steady state solution for a linear network is to start with the state equations of the circuit:

$$\underline{\dot{x}} = A\underline{x} + Bu \tag{4}$$

$$y = C\underline{x} + du \tag{5}$$

Then, through the transfer function

$$T(s) = d \frac{\det[sI - (A - \frac{1}{d}BC)]}{\det[sI - A]} \dots, d \neq 0 \tag{6}$$

we can find the poles and zeroes of $T(s)$, and evaluate $T(s)$ at each different frequency.

For time-domain solution, we can obtain the partial fraction expansion of $T(s)$, and then by the inverse Laplace transform we can find the solution directly.

Recently there is a method²¹ to obtain transfer function with fast speed and numerical accuracy

$$= A\underline{x} + Bu \tag{7}$$

$$y = C\underline{x} \tag{8}$$

The procedures are

- (1) Using double QP transformation technique to find the eigenvalues of A -the poles of $T(s)$
- (2) To find the zeros of $T(s)$ between a output y_r and an input u_s is as follows:
 - (a) find k so that $|C_k^r| \geq |C_i^r|, i=1,2,\dots,n$
 - (b) Determine $TAT^{-1} \triangleq (a_{ij}); TD_s \triangleq (b_1, b_2, \dots, b_n)$ where

$$T = \begin{bmatrix} I_{k-1} & 0 \\ C_k^r & 0 \\ 0 & I_{n-k} \end{bmatrix}$$

- (3) Choose k^* to be an arbitrary large number compared to the magnitude of the numbers $|a_{ij}|; i=1,2,\dots,n; j=1,2,\dots,n$, so that $\max_i |b_i| k^{**} \gg \max_{i,j} |a_{ij}|$,

$$(\text{say } k^* = 10^{15} \frac{\max_{i,j} |a_{ij}|}{\max_i |b_i|})$$

(4) Find the eigenvalues of the matrix A where

$$A = \begin{bmatrix} a_{11} \dots a_{1,k-1} & k^*b_1 & a_{1,k+1} \dots a_{1n} \\ a_{21} & a_{2,k-1} & k^*b_2 & a_{2,k+1} \dots a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n,k+1} & k^*b_n & a_{n,k+1} \dots a_{nn} \end{bmatrix}$$

(5) Reject the eigenvalues which approach to ∞ . The zeros of the system are equal to the remained eigenvalues.

For finding an optimum circuit to meet a given specification, we repeatedly analyze a trial circuit by adjusting the circuit parameters in a systematic way. The successful parameter optimization of a circuit depends on the efficient computation of the gradient vector of the desired circuit function. The successful parameter optimization can be obtained by the following steps:

- Step 1: By state variable formulation method and network partition technique to obtain hybrid state equations [2]. The set of equations contains only state variables and parameter variables, and parameter adjustments can be made effectively in the set of hybrid state equations.
- Step 2: By finding adjoint network [12] to obtain sensitivity vector of performance index to parameters. Because it is easy to construct the adjoint network from the original network and moreover the complexity of adjoint network is same as original network. So, we can make the gradient vector of performance to parameters easily obtainable.
- Step 3: By the Fletcher-Powell [1] to find the optimum parameters. The Fletcher-Powell minimization procedure is a suitable optimization method for our scheme. Because it combines the best feature of linear and quadratic convergence methods and use the knowledge of performance index and its gradient vector only.

7. PROGRAM FOR LAYOUT

Once circuit design is complete, it remains to layout printed circuit. The layout design relies heavily on designer's intervention on component locating and subsequent interconnection routing.

The basic method of the locating algorithm is a modification of the idea of ACCEL¹⁴. There are five types of forces acting on each component, such as attractive or repulsive force between components, moment of rotation, etc. The total force acting on a component will shift the component to its equivalent position.

The routing algorithm in our system is a modification of Lee's¹⁵, in which the merit of the heuristic approach is included. A three-dimensional model is first set up for the circuit. The best path for connecting two lands, which are the locations of the leads of component, is along a plain or valley.

The locating and routing algorithms must be integrated to satisfy the criterion of minimum interconnection wire length¹⁶. This can be accomplished by the iterative method as shown in Fig. 6.

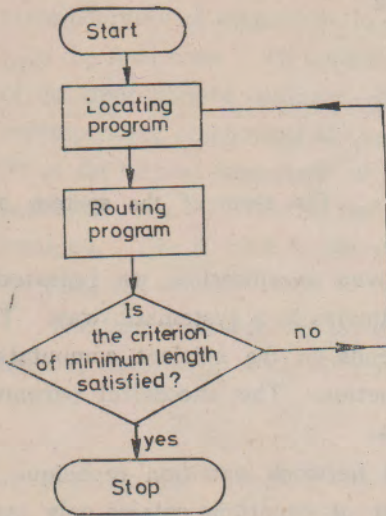


Fig. 6. The Flowchart of Component Placements and Interconnection

For computer-aided mask artwork, we had better store in the disk or tape of the central computer an extensive library of component types and their physical realizations.

8. CONCLUDING REMARK

An overall view of the on-line circuit design system to be developed at Chiao-Tung University has been presented. So far, the circuit analysis program, and the component locating and routing programs were completed. However, the modifications of these programs to DDC/10 are still needed. In the second year project, we will complete the whole hardware structure, and will concentrate on designing an input-output controlling program and an user-oriented input language.

ACKNOWLEDGEMENT

It is our deep gratitude to thank to National Science Council for supporting this project.

REFERENCES

1. F.F. Kuo & W.G. Manguson, "Computer Oriented Circuit Design," 1970.
2. Y.S. Huang, "Electronic Circuit Design Program," The M.S. Thesis, National Chiao-Tung University, 1972.
3. B.F. Jiang, "Computer-Aided Printed Circuit Board Design," The M.S. Thesis, Chiao-Tung University, 1973.
4. H.C. So, "Aspects of Computer-Aided Circuit Design," Technical Report, Bell Telephone Lab., Murray Hill, N.J.
5. C. Chen, "A Study of On-Line Computer System for Circuit Design," Technical Report, Chiao-Tung University, 1973.
6. H.C. So, ROLCA: "On-Line Circuit Analysis System," Proc. of IEEE, Vol. 55, No. 11, pp 1954-1964, 1967.
7. M.L. Dertouzos, "An Introduction to On-Line Circuit Design," Proc. of IEEE, Vol. 55, No. 11, pp 1964-1971, 1967.
8. M.L. Dertouzos, "CIRCAL: On-Line Circuit Design," Proc. of IEEE, Vol. 55, No. 5 pp 637-653, 1967.
9. D.L. Dietmeyer & Y.H. Su, "Logic Design Automation of Fan-In Limited NAND Networks," IEEE Tran. on Computers, Vol. C-18, Jan. 1969.
10. Semionual Technical Summary-Graphic, M.I.T. Lincoln Lab., Lexington, Mass., Report AD 67999, Nov. 1968.
11. C.W. Gear, "The Numerical Integration of Stiff Ordinary Differential Equations," Department of Computing Science, University of Illinois, Urbana, Ill., Report 221, Jan 1967.
12. P. Penfield, P. Spence, & S. Duinker, "A Generalized Form of Tellegen's Theorem," IEEE Trans. on CT, Vol. CT-17, Aug. 1970.
13. H.C. So, "Network Partitions and A. General Sensitivity Formula," Technical Report, Bell Telephone Lab., Murray Hill, N.J.

14. C.J. Fisk, & D.L. Caskey, "ACCEL: Automated Circuia Card Etching Layout, "Proc. of IEEE, Vol. 55, No. 11, Nov., 1967.
15. C.Y. Lee, "An Algorithm for Path Connections and Ins Application," IRE Trans. Electronic Computer, Vol. EC-10, Sept., 1961.
16. F.G. Bowden & G.J. Lovitt, "Automatic Layout of Printed Board," Electrical Communication, Vol. 47, No. 4, 1972.
17. M.A. Breuer. "General Survey of Design Automation of Digital Computer," Proc. of IEEE, Vol. 54, No. 12, pp 1708-1721, Dec. 1966.
18. G. Gorden, "A General Purpose Systems Simulation Program," 1961 Proc. EJCC, pp 87-104.
19. D.E. Knuth and J.L. McNeley, "SOL-A Symbolic Language for General Purpose Systems Simulation," IEEE Trans. on Electronic Computer, Vol. EC-13, pp 401-403, Aug. 1964.
20. D.S. Evans and J. Katzenelson, "Data Structure and Man-Machine Communication for Network Problems," Proc. of IEEE, Vol. 55, No. 7, pp 1135-1144, July 1967.
21. B.J. Davison, "On the Calculation of Zeros of A Linear Constant System," IEEE Trans. on Circuit Theory, Vol. CT-18, pp 183-184, Jan. 1971.
22. R.D. Berry, "An Optimal Ordering of Electronic Circuit Equations for a Sparse Matrix Solution," IEEE Trans. on Circuit Theory, Vol. CT-18, No. 1, pp 40-50, Jan. 1971.
23. T.E. Stern, "On the Equations of Nonlinear Network," IEEE Trans. on Gircuit Theory, Vol. CT-13, pp 74-81, Mar. 1966.