

no. 14

**Design  
Automation  
Conference  
PROCEEDINGS**

**June 20-22, 1977**



IEEE COMPUTER  
SOCIETY-DATC



# **NO.**

# **14**

# **Design Automation Conference**

## **Proceedings**

**June 20, 21 & 22, 1987 New Orleans, Louisiana**

IEEE Catalog Number 77 CH1216-1C

The ideas and opinions expressed herein are solely those of the Authors and are not necessarily representative of or endorsed by the 1977 Design Automation Conference Committee, the Association for Computing Machinery or The Institute of Electrical and Electronics Engineers, Inc.

IEEE Catalog Number 77 CH 1216-1C  
Library of Congress Catalog Card No. 74-30514

Copyright © 1977 by The Institute of Electrical and Electronics Engineers, Inc., 345 East 47 St., New York, N. Y. 10017, and the Association for Computing Machinery, 1133 Avenue of the Americas, New York, N.Y. 10036.



Welcome to New Orleans and the 14th Design Automation Conference. This year we have 70 technical papers covering new innovations and techniques in the automation of various aspects of electrical, mechanical, and architectural design. In addition, we have a number of sessions with a tutorial-type purpose to give to those new in the field an overview of the techniques employed in a particular area of design automation. And, of course, we will continue to have the exciting "Round Table" sessions in which these well-known





## Chairman's Introduction

Welcome to New Orleans and the 14th Design Automation Conference. This year we have 70 technical papers covering new innovations and techniques in the automation of various aspects of electrical, mechanical, and architectural design. In addition, we start each conference day with a tutorial whose purpose is to give to those new in the field an overview of the techniques employed in a particular area of design automation. And, of course, we will continue to have the evening "birds of a feather" sessions to allow those with common interests in a specific area to get together for continued, in-depth technical discussions -- as in the past, we hope that conferees who would like to suggest a particular topic for a "birds of a feather" session will coordinate with someone at the registration desk to get a sign-up sheet started.

Why did you come to the Design Automation Conference this year? Some of you are regulars -- we see you every year. And, if past statistics hold up, about a third of you are new -- either new to the DA field or just first-time participants in this particular conference. And, of course, there are a few who searched the conference listings for a nice vacation spot, and chose New Orleans this year. But, seriously, why do we come to the conference?

Speaking for myself, I'd like to say that I've been attending these conferences for seven years now and I believe that I have genuinely benefited from the interactions here. What happens to all of us, I believe, is that we get so enmeshed in our own job and our own way of doing things -- whether it be how we attack the design of an algorithm, the structuring of a new system or the management of a project -- that we rarely take the time to stand back and reevaluate our methods. When we come to the Design Automation Conference we get away for three days -- away from the stack of mail and the telephone and the pressures of immediate needs. While we are here we are still thinking about our jobs, but usually we are thinking about them with respect to the views of the speaker at a technical session, a counterpart during a coffee break or dinner, or the discussion going on at a "birds of a feather" session. Technically, even managerially, our problems are very much the same, and yet we sometimes handle these problems in very different ways. I say, sometimes, because actually I find that for the greatest part, even though we come from companies all over the world, our method of handling design automation problems are remarkably alike. But it is the differences that stand out and are remembered. And, personally, I always go back to my job with a new idea and a renewed desire to approach my job in a different manner.

This year, we have added two new items to our conference, a Best Paper Award and an Artwork Show and Contest. With the Best Paper Award, we hope to make special recognition of the tremendous effort required to produce a top-notch paper that presents some kind of breakthrough in the design automation field in a way that makes it easily understandable and interesting to the reader. Our method of handling this award was to ask the two technical reviewers of each paper to indicate if the paper was of particular high quality. Of the 70 papers selected for the conference, 8 of these papers were designated by reviewers as being of particular merit. A committee of four was then appointed and each member of the committee was asked to review all eight papers in terms of this Best Paper Award and list them in the order of their preference. The top four papers from these votes based on a direct comparison of the original eight papers are the finalists for the award and each of you has a ballot, allowing you sometime during Monday or Tuesday to read these four outstanding papers, and vote on it by placing your marked ballot in the box at the registration desk. The award will be made during the Wednesday luncheon by Steve Chappel, who suggested the idea of having the Best Paper Award. During the conference, we would appreciate it if you would make your feelings and suggestions concerning the method of handling future DAC Best Paper Awards known to any DAC Committee Member.

The Artwork Show and Contest has also been added to the Conference this year to allow as a means for viewing useful output of each others design automation system with the contest part included to generate interest. In this case, the choice will be made by a New Orleans artist. The award will be presented at the Wednesday luncheon by Paul Losleben, who suggested this addition to our conference.

The DAC Committee works as a team to organize and hold this conference. As the conference has grown in size and scope, the organization problem has grown with it. This is an extracurricular activity that requires a great deal of effort from people who are already busy. I would like to express my appreciation to the members of the committee and to those in our sponsoring organizations -- the ACM's Special Interest Group in Design Automation and the IEEE Computer Society's Design Automation Technical Committee who have made this conference possible. Working with these people over the last year has been both a rewarding and pleasurable experience. But you, the conferees, are also responsible for making the conference a success by your contributed papers, discussions, and interactions in the next three days so let's all get going at our continued effort to make DAC '77 an outstanding conference.

J. G. Brinsfield

# PROCEEDINGS OF THE DESIGN AUTOMATION CONFERENCE

1977 DESIGN AUTOMATION  
CONFERENCE COMMITTEE

General Chairman  
J. B. Burchfield  
Bell Laboratories  
Building 2B-311  
Holmdel, New Jersey 07733

2ND	(1965)	Contains 20 Papers	360 Pages
3RD	(1966)	Contains 16 Papers	410 Pages
4TH	(1967)	Contains 24 Papers	500 Pages
5TH	(1968)	Contains 28 Papers	580 Pages
6TH	(1969)	Contains 33 Papers	414 Pages
7TH	(1970)	Contains 36 Papers	345 Pages
8TH	(1971)	Contains 40 Papers	383 Pages
9TH	(1972)	Contains 47 Papers	375 Pages
10TH	(1973)	Contains 36 Papers	288 Pages
11TH	(1974)	Contains 47 Papers	379 Pages
12TH	(1975)	Contains 57 Papers	448 Pages
13TH	(1976)	Contains 63 Papers	501 Pages

Dallas, Texas 75222  
214-388-6666

Finance Chairman  
F. O. Fradin

Copies of the Proceedings of past Conference meetings can be obtained from either of the following organizations.

ACM  
Order Department  
P. O. Box 12105  
Church Street Station  
New York, New York 10249

IEEE Computer Society  
ATTN: Mr. True Seaborn  
5855 Naples Plaza  
Suite 301  
Long Beach, California 90803

Either of these organizations should be contacted before placing your order to determine availability of past proceedings of individual meetings.

Rowin B. Rasmus  
Team Instruments Incorporated  
P.O. Box 5012, M.S. 307  
Dallas, Texas 75222  
214-388-6666

Arrangements Chairman

Donald J. Finnegan

Bell Laboratories

Building 1F-214

Holmdel, New Jersey 07733

201-949-6328

ACM Representative

Martin J. Well

Bell Laboratories

Building 2B-311

Holmdel, New Jersey 07733

201-949-6328

IEEE Representative

Roy L. Huns

IBM T. J. Watson Research Center

P.O. Box 218

Yorktown Heights, New York 10598

914-945-1848



# 1977 DESIGN AUTOMATION CONFERENCE COMMITTEE

## General Chairman



Judith G. Brinsfield  
Bell Laboratories  
Building 3B-323 Whippany Rd.  
Whippany, New Jersey 07981  
201-386-3169

## Vice-Chairman



Stephen A. Szygenda  
University of Texas  
Electrical Engineering Dept. (ENS515)  
Austin, Texas 78712  
512-471-7365

## Program Chairman



David W. Hightower  
Texas Instruments Incorporated  
P.O. Box 5012, M.S. 907  
Dallas, Texas 75222  
214-238-5667

## Finance Chairman



Pat O. Pistilli  
Bell Laboratories  
Bldg. 1F-08  
11900 N. Pecos St.  
Denver, Colorado 80234  
303-451-4364

## Publicity Chairman



Robert B. Hitchcock  
IBM Dept. V41 Bldg. 028-4  
1701 North Street  
Endicott, N.Y. 13760  
607-755-7853

## Publications Chairman



Edwin B. Hassler  
Texas Instruments Incorporated  
P.O. Box 5012, M.S. 907  
Dallas, Texas 75222  
214-238-5666

## Arrangements Chairman



Donald J. Humcke  
Bell Laboratories  
Building 1F-214  
Holmdel, New Jersey 07733  
201-949-6253

## ACM Representative



Martin J. Welt  
Bell Laboratories  
Building 2B-311  
Holmdel, New Jersey 07733  
201-949-5588

## IEEE Representative



Roy L. Russo  
IBM, T. J. Watson Research Center  
P.O. Box 218  
Yorktown Heights, New York 10598  
914-945-1643



PROCEEDINGS OF THE  
FOURTEENTH ANNUAL DESIGN AUTOMATION CONFERENCE  
UNDER JOINT  
SPONSORSHIP OF THE  
ACM AND IEEE

Chairman's Introduction	J. G. Brinsfield	
<b>Session 1 INTERCONNECTION</b>		
Chairman	L. C. Abel	
1. A Minicomputerized Automatic Layout System for Two-Layer Printed Wiring Boards	I. Nishioka, T. Kurimoto H. Nishida, I. Shirakawa, H. Ozaki	1
2. A Multi-Contouring Algorithm	I. Dobes	12
3. A Rectangle-Probe Router for Multilayer P. C. Boards	W. G. Cage R. J. Smith, II	13
4. Some Theoretical Aspects of Algorithmic Routing	P. Agrawal M. A. Breuer	23
5. Prediction of Wiring Space Requirements for LSI	W. R. Heller, W. F. Mikhail W. E. Donath	32
6. Computer/Interactive Cleanup of Non-Gridded PWB's After Automatic Routing	D. P. Peterson	43
<b>Session 2 TESTING</b>		
Chairman	C. Radke	
7. A New Look at Test Generation and Verification	P. G. Kovijanac	58
8. An Automated Simultaneous Probing System for Testing Complex Logic Assemblies 'The Bed of Nails System'	R. H. Somaia	64
9. Computer Aided Test Pattern Generation for Digital Processors	K. Pfeuffer	68
10. Automatic Test Generation for Large Digital Circuits	A. Yamada, N. Wakatsuki, H. Shibano O. Itoh, K. Tomita, S. Funatsu	78
11. Heuristic Enhancement of an Algorithmic Test Generator	R. E. Strebendt	84
12. Simulator-Oriented Fault Test Generator	T. J. Snethen	88
<b>Session 3 MECHANICAL DESIGN AUTOMATION</b>		
Chairman	P. Kennicott	
13. Tools for Map Graphics	P. Fulton	94
14. Flight Test Analysis of Missile Control Systems	M. Domaszewicz	101
15. MIDAS An On-Line Real Time Material System	R. S. Hall	109
16. Using a Computer Aided Graphics System to Help Design and Draft Automotive Components	N. L. Brown	112
<b>Session 4 FAULT TOLERANT DESIGN AUTOMATION</b>		
Chairman	C. Rose	
17. Fault Modeling in a Hierarchical Simulator	J. J. Strunge	118

18. Concurrent Fault Simulation and Functional Level Modeling

M. Abramovici, M. A. Breuer  
K. Kumar

128

## Session 5 IC LAYOUT

Chairman

W. M. vanCleemput

19. FLOSS: An Approach to Automated Layout for High-Volume Designs
20. Analytical Power/Timing Optimization Technique for Digital System
21. An Experimental System for Power/Timing Optimization of LSI Chips
22. The Siemens-AVESTA-System for Computer-Aided Design of MOS-Standard Cell Circuits
23. Automated ECL LSI Design

Y. E. Cho, A. J. Kroenjak,  
D. E. Stockton

138

A. E. Ruehli, P. K. Wolff, Sr.,  
G. Goertzel

142

B. J. Agule, J. D. Lesser,  
A. E. Ruehli, P. K. Wolff, Sr.

147

K. W. Koller,  
U. Lauther

153

N. R. Crocker, R. W. McGuffin,  
A. Mickelthwaite

158

## Session 6 PWB LAYOUT

Chairman

B. Heller

24. A Production PCB Layout System on a Minicomputer
25. DOCIL: An Automatic System for Printed Circuit Board (PCB) Designing. A Board Description Language and an Algorithm to Connect a Set of Points.
26. A Human Engineered PCB Design System

K. Bedard, S. Fournier,  
B. Shastry, U. Stockburger

168

M. T. de Pedro  
R. Garcia

174

A. J. Matthews

182

## Session 7 SIMULATION

Chairman

E. Thompson

27. A Concept for the Editing of Hardware Resulting in an Automatic Hardware-Editor
28. Simulation of Large Communications Networks Using SPIN
29. Logic Design Automation of Diagnosable MOS Combinational Logic Networks
30. Practical Experiences from Signal Probability Simulation of Digital Designs
31. Detection of Static and Dynamic Hazards in Logic Nets
32. Simulation Techniques for Microprocessors
33. An Efficient Method of Fault Simulation for Digital Circuits Modeled from Boolean Gates and Memories

F. J. Rammig

187

I. L. Morris, J. McNulty,  
R. Gee

194

Y. M. El-Ziq  
S. Y. H. Su

205

B. Magnhagen

216

A. K. Bose,  
S. A. Zyngenda

220

J. R. Armstrong, G. Woodruff

225

D. M. Schuler,  
R. K. Cleghorn

230

## Session 8 ARCHITECTURAL DESIGN AUTOMATION

Chairman

R. Frew

34. Uncertainty and Optimization in the Design of Building Subsystems
35. Symbols, Graphics and Architectural Education: The Pagan Experience
36. An Affordable Approach to an Architectural Computer System
37. Evolution of a Spatial Allocation System: ALLOCATE

E. E. Dudnik

239

M. Kennedy

244

D. E. Bergeson  
R. Babbitt

254

B. Jackson

265



38. The Site Machine: Computer-Aided Instruction In Architectural Education	E. F. Smith	266
39. Computer Aided Design in North American Schools of Architecture	R. S. Frew	275
College of Architecture, University of Arizona	R. W. Dvorak	276
Computer-Aided Design and Practice in City College School of Architecture	G. A. Gebert	277
Computer-Aided Architectural Design	E. Teicholz	279
Department of Architecture University of Illinois	K. E. Tanaka	280
Computer Aided Design, College of Architecture, University of Kentucky	M. Kennedy	281
A Second Chance at Automation as a Design Tool	R. J. Hogan	282
Yale School of Architecture	R. S. Frew	283
<b>Session 9 PLACEMENT &amp; WIRING</b>		
Chairman	P. Wolff	
40. A Class of Min-Cut Placement Algorithms	M. A. Breuer	284
41. The Chip Layout Problem: A Placement Procedure for LSI	K. H. Khokhani, A. M. Patel	291
42. The Chip Layout Problem: An Automatic Wiring Procedure	K. A. Chen, M. Feuer, K. H. Khokhani, N. Nan, S. Schmidt	298
<b>Session 10 DESIGN RULE VERIFICATION</b>		
Chairman	R. Mattison	
43. Fast Algorithms for LSI Artwork Analysis	H. S. Baird	303
44. A Comprehensive Approach to a Connectivity Audit, or a Fruitful Comparison of Apples and Oranges	R. M. Allgair, D. S. Evans	312
45. A Layout Checking System for Large Scale Integrated Circuits	K. Yoshida, T. Mitsuhashi, Y. Nakada, T. Chiba, K. Ogita, S. Nakatsuka	322
<b>Session 11 USER EXPERIENCES IN DESIGN AUTOMATION</b>		
Chairman	L. Marks	
46. The Open Shop Interactive Mask Design Operation at Harris Semiconductor	J. E. Harlow, III	331
47. Automatic Optical Design with ACCOS V Program	M. Amon	336
48. A Design Automation System for Printed Circuit Board Assemblies	A. Bobas, J. Valihora	341
49. Computer Designed Multilayer Hybrid Substrate Using Thick Film Technology	C. W. Waldvogel	351
<b>Session 12 DESIGN VERIFICATION</b>		
Chairman	D. Schweikert	
50. Design Verification of Large Scientific Computers	H. E. Krohn	354
51. A Design Verification and Logic Validation System	W. A. Noon	362
52. Designing with LCD: Language for Computer Design	C. J. Evangelisti, G. Goertzel, H. Ofek	369
53. An Hierarchical Language for the Structural Description of Digital Systems	W. M. vanCleeemput	377



## Session 13 DOCUMENTATION

Chairman

54. Cost Effective Layout Digitizing and Mask Pen Plotting of Custom Microelectronic Devices
55. Automating Analog Circuit Diagrams Using A List Processing Language
56. CASS: Computer Aided Schematic System (A powerful and practical design aid module in an integrated data-base oriented CAD environment)
57. Manipulation of Design Data

D. L. Peterson

R. P. Larsen

R. C. Jaffe,  
J. P. Young

H. M. Bayegan

C. A. Linden

386

391

396

405

## Session 14, DESIGN AUTOMATION SYSTEMS

Chairman

58. Measuring Designer Performance to Verify Design Automation Systems
59. A Design Automation System for Telephone Electronic Switching System
60. SWESS - The Middle System of a Design Automation Network
61. SPIDER - A Computer Aided Manufacturing Network
62. Piramed Project An Integrated CAD/CAM System Development
63. CDL - A Tool for Concurrent Hardware and Software Development?
64. Thick Film Substrate (Micropackage) Design Utilizing Interactive Computer Aided Design Systems

J. J. Morzenti

D. E. Thomas,  
D. P. Siewiorek

F. E. Swiatek

D. K. Carley

M. E. Walsh

R. W. Srch

J. R. Heath, B. D. Carroll  
T. T. Cwik

F. M. Christley

411

419

425

431

437

445

450

## Session 15 AN LSI TEST SYSTEM

Chairman

65. Introduction to an LSI Test System
66. A Logic Design Structure for LSI Testability
67. Automatic Checking of Logic Design Structures for Compliance with Testability Ground Rules
68. Test Generation for Large Logic Networks
69. Delay Test Generation
70. Delay Test Simulation

M. Correia

M. Correia, F. B. Petrini

E. B. Eichelberger,  
T. W. Williams

H. C. Godoy, G. B. Franklin,  
P. S. Bottorff

P. S. Bottorff, R. E. France,  
N. H. Garges, E. J. Orosz

E. P. Hsieh, R. A. Rasmussen,  
L. J. Vidunas, W. T. Davis

T. M. Storey, J. W. Barry

R. J. Smith, II

460

462

469

479

486

492

495

## A TUTORIAL

Software Engineering Techniques in Design Automation

R. J. Smith, II

495

# A MINICOMPUTERIZED AUTOMATIC LAYOUT SYSTEM FOR TWO-LAYER PRINTED WIRING BOARDS

Ikuro Nishioka, Takuji Kurimoto, and Hisao Nishida

Central Research Laboratory, Sharp Co., Ichinomoto, Tenri, Nara, 632 Japan

Isao Shirakawa and Hiroshi Ozaki

Department of Electronic Engineering, Osaka University, Suita, Osaka, 565 Japan

## Abstract

The requirement for high density packaging is dominant in the design of electronic systems. In the assembly of such systems multi-layer printed wiring boards are often used to provide the necessary connections among functional modules, and as the wiring density for a board ascends, an automatic scheme to realize 100 percent wiring would be more essential to reduce cost and time incurred in laying out wire pattern. Thus, methods to improve the layout of printed wiring boards are still continually under investigation.

The present paper proposes an automatic layout system for two-layer printed wiring boards, which operates on a PDP 11/40 computer with 24K 16-bit words of core memory and with 1.2M words of disk storage, coupled with a TEKTRONIX 4014 display terminal in conjunction with a 4953 graphics tablet. This system has been at work, and more than 300 wiring boards of manufacturing use have been already processed automatically by this system, almost all with 100 percent routing realized. A part of such implemented results are also shown.

## 1. Introduction

In the design of electronic systems, the high density packaging is one of the most important objectives in order to reduce cost and improve reliability. In the assembly of such systems two-layer printed wiring boards are still most used to provide the necessary connections among functional modules, and hence of practical importance is the layout scheme to realize high density wiring on such boards.

Generally, the circuit modules to be interconnected by means of a two-layer wiring board are mounted on the first side of the board, each with connector pins inserted through conducting holes, called plated-through holes, so that each set of the pins to be made electrically common can be connected by conductor paths, or wire routes, composed of horizontal and vertical segments on the first and second sides, respectively, and vias to connect segments on different sides. Thus, the process of layout for such a wiring board is to be divided into two phases: The first is the placement of circuits modules on the first side, and the second is the routing of conductor paths on both sides. To these problems a variety of approaches have been proposed and extensively developed (see, for example, excellent literatures [1-3]). However, especially as to the second problem, in most instances routing algorithms are non-iterative, that is, wires are laid out one at a time in a specified order, and meanwhile once they are laid out, they can not be modified later. Hence, it is essential that wires be routed in the 'optimum order' subject to some criterion. As is often employed for two-layer wiring boards, let the performance of a router be measured by the number of unrouted wires, then the ordering scheme is very critical [4,5], but any of schemes so far proposed is not fully satisfactory to this end. Thus, there is a tendency that as the wiring density for a board ascends, the flexibility, and therefore wirability, of any router based on non-iterative algorithms descends. To cope with this, new approaches are still continually under investigation [5,6]. On the other hand, as another

way out of this difficulty, of practical use is the interactive layout scheme based on a minicomputer (for example, see [7]).

The present paper proposes a minicomputerized layout system for two-layer printed wiring boards, which operates on a PDP 11/40 computer coupled with a TEKTRONIX 4014 display terminal in conjunction with a 4953 graphics tablet. This system is distinctive mainly in that the routing scheme, composed of line-search and maze-running routines, is automatic in the following sense; a user who is a little acquainted with specific features of the two routines, has only to try to modify the current wire pattern so as to make an open space in order that each wire route still lacking can be laid out automatically by applying any of the two routines.

This system has been in use, and is contributing not a little toward reducing cost and time incurred in manufacturing printed wiring boards. To observe its performance, a number of results so far implemented for practical purposes are also outlined.

## 2. Hardware Configuration

An outline of the hardware configuration of the system is shown in Fig. 1: This system operates on a PDP 11/40 computer with 24K 16-bit words of core memory and with 1.2M words of disk storage, coupled with a TEKTRONIX 4014 display terminal in conjunction with a 4953 graphics tablet. In addition, the system is equipped with a card reader, paper tape reader/punch, 9 track magnetic tape unit, and a console typewriter. A digitizer is used off-line to provide input data concerning pin locations of each circuit module on the board. A drum-plotter is used for check drawing of wire pattern generated by the routing process, and a photo-plotter is for photo realization of wire pattern completed in Artwork File.

## 3. Software Configuration

The software configuration of the system is outlined as shown in Fig. 2, where each routine has a function summarized as in Table 1.

Table 1 Function of Each Routine

Routine Name	Function
GET-S	To read input data and to construct the source data file
EDIT-G	To correct those errors, which are detected as the input is processed, directly at the graphics display without undue delay caused by paper tape or card handling
G-FILE	To build the design data base from the source data file
M-FILE	To manage the design data base
LINE	Line-search routine
MAZE	Maze-running routine
S-GRAPH	Interactive editing routine for the design data base
C-GRAPH	Interactive editing routine for the artwork file
NC Tapes Generating A	To generate NC tapes to be inputted to a drum-plotter for check drawing
NC Tapes Generating B	To generate NC tapes to be inputted to a photo-plotter and a drilling machine



S-CHECK	To check short- and disconnection-errors of wire pattern which may possibly arise in executing S-GRAPH
C-CHECK	Closeness checking for wire pattern in executing C-GRAPH
PLACE	Replacement routine

Note that in this system the design data base is accessed by pairwise independent routines such as

1. Line-search routine (LINE),
2. Maze-running routine (MAZE),
3. Interactive editing routine (S-GRAPH),
4. NC tapes generating routine (NC Tapes Generating A),
5. Checking routine (S-CHECK),
6. Replacement routine (PLACE).

Here, the line-search routine (LINE) and maze-running routine (MAZE) constitute the routing scheme of this system, which is outlined later. The interactive editing routine (S-GRAPH) is for modifying the current wire pattern with the aid of a CRT display system so that each wire route still lacking can be laid out automatically by applying any of the two routing routines, as is discussed later. The NC tapes generating routine (NC Tapes Generating A) is to produce magnetic tapes to be inputted to a CALCOMP 1036 drum-plotter and for back-up store. The checking routine (S-CHECK) is for checking automatically any short among wire routes associated with different signal nets\* and disconnection of wire routes associated with the same net, which may possibly arise in executing modifications of wire pattern with the use of the CRT display in S-GRAPH. In addition, the replacement routine (PLACE) is optionally provided to modify the initial placement of the modules on the board, with the use of the CRT display, which is also outlined later.

#### 4. Implementation Overview

Normally, a design originates in the form of providing input data to describe the pin locations of circuit modules on the board and the net interconnection. As the input is processed, numerous error conditions, are detected, and these errors, if any, may be corrected directly at the graphics display without the undue delay caused by paper tape or card handling. Once the input has been read, the design data base is constructed, and the system waits next operations.

As to implementing the routing scheme, in its first run, the process is fully automatic with the use of the net ordering scheme, to be stated later, and begins with the line-search routine for all nets in the specified order. Then, for those nets which are still unconnected, the maze-running routine is executed in the order specified before.

At a stage when the user gets 100 percent routing and is satisfied with the wire pattern, he executes checking the net interconnection on a drum-plotter, then shifts the operation into the second stage around Artwork File, which is constructed from the design data base. Otherwise, the graphics editing system for wire pattern plays an active part. The user may be devoted to updating the current wire pattern in order to make isolated pins be connected in the next run of the routing process.

For this purpose, the interactive updating routine in this system offers various kinds of instructions which will be discussed later. With the use of these interactions, the wire routes still missing in the current pattern are to be laid out at a time in the next run of automatic routing, as in the following policy:

\* A signal net indicates a set of the pins to be made electrically common.

For each net still unconnected, a user first tries to modify the current wire pattern so as to make an open space such that the wire route between an isolated pin and the connected part of the net can be generated through the open space, and he then restarts the routing process, ordinarily the maze-running routine. Series of graphics updating and automatic routing can be repeated until the user is satisfied with the wire pattern so far generated.

Thus, in this system any new wire route is generated in principle by automatic routing rather than manual routing, which might be so complicated as to be liable to make wiring errors, as well as to be time-consuming, especially when the wiring density is high.

Now, we must also pay attention to the fact that we are confronted very often with various kinds of unstandardized modules to be mounted on the board. In such a case, it is desirable that a user can execute graphics updating, observing a realistic artwork pattern on the screen. To this end, we provide a powerful tool named C-GRAPH, as will be stated later.

Once a wire pattern layout of a board is completed, facilities are available through this system to obtain a full array of design documentation. This documentation includes artmasters for wiring board fabrication, check drawing, and NC tapes generation for drilling machine and back-up store.

#### 5. Board Specification

The effective area for wiring use is divided into columns and rows, both with the same width 1.27mm (or 1/20 inch), which is the half of the interval between two consecutive pins of ordinary DIP IC's, and let a square region located in column  $x$  and row  $y$  be designated as a cell  $(x,y)$ . Henceforth, let it be supposed, unless otherwise stated, that

- 1° such IC's are placed on the board with dual pin-lines oriented vertically,
- 2° horizontal (vertical) wire segments lie in rows on the first side (in columns on the second side),
- 3° any plated-through hole or via occupies a cell, under the condition that any two must not be located at pairwise adjacent cells, where cell  $(x,y)$  is said to be adjacent to cell  $(x',y')$  if and only if  $|x-x'| + |y-y'| = 1$ .

The wiring area of 512 rows by 512 columns is the maximum size applicable to this system, for which 32,768 words (=524,288 bits) are provided to form the bit map necessary in the line-search routine, with 8,192 words (equivalently, 256 rows by 256 columns) in core and excess words in disk.

#### 6. Routing Scheme

A line-search algorithm and a maze-running algorithm constitute the routing scheme of this system, and any of these is to be conducted for some specified signal nets, one at a time, in a predetermined order by ordering scheme, or for a single net indicated by a user, in such a manner that in the routing process for a net the disk is not accessed, but once the process halts for the net, all the necessary informations about the wire segments created in this process are transferred from the core to the disk.

##### A. Line-Search Algorithm

In the following, an outline of the line-search algorithm adopted in this system is described, by which usually 80-90 percent routing can be attained. The line-search method, originally proposed by Mikami-Tabuchi [8] and extensively used, consists essentially in finding a wire route by searching for those candi-



dates which may possibly be chosen as horizontal and vertical line segments of the wire route. The algorithm, based on Yamamura-Shirakawa-Ozaki [9], is so sophisticated as to be implemented by a minicomputer.

Throughout the routing process, each signal net is partitioned into two; the one is a set of the isolated pins and the other is a set of pairwise connected pins, and associated with an incomplete net, routing procedure is, at any stage, devoted to creating a wire route to connect each isolated pin to the connected part, that is, the connected wire pattern so far generated for the net.

To describe the line-search algorithm, we need some definitions as follows: Associated with a pin  $t_i$  located at cell  $(x_i, y_i)$ , let us denote by a segment of level 0, written by  $\ell_0(t_i)$ , a possible wire segment candidate in one of the following shapes

- 1° a maximal horizontal line segment to be newly drawn on the first side from cell  $(x_i, y_i)$ ,
- 2° a vertical segment from  $(x_i, y_i)$  to unoccupied  $(x_i, y_i \pm 1)$  and a maximal horizontal one to be newly drawn from  $(x_i, y_i \pm 1)$ , both on the first side.

A segment of level 1, written by  $\ell_1(t_i)$ , indicates a possible wire segment candidate in the shape of a maximal vertical line segment to be newly drawn on the second side from a cell located on any  $\ell_0(t_i)$ , and a segment of level 2, written by  $\ell_2(t_i)$ , a possible wire segment candidate in the shape of a maximal horizontal line segment to be newly drawn on the first side from a cell located on  $\ell_1(t_i)$ , which does not coincide with any  $\ell_0(t_i)$ .

A rough sketch of the line-search algorithm is as follows: Let  $T = \{t_1, t_2, \dots, t_n\}$  be a signal net to which the algorithm is to be applied, and let US and CS denote the unconnected set, that is, a set of the isolated pins, and the connected set, that is, a set of the pairwise connected pins, of the net, respectively, where it is noted that  $T = US \cup CS$ ,  $US \cap CS = \emptyset$ .

The algorithm consists essentially in finding those wire routes to connect each  $t_i \in US$  to some of the wire segments, already settled, connecting pins in CS, which are limited to be composed of three or less horizontal wire segments and two or less vertical segments. This limitation on generating wire pattern aims at speeding up the search.

Initially (when  $CS = \emptyset$ ), the algorithm seeks an interval  $[x_1, x_2]$ , over which segments  $\ell_0(t_i)$  of level 0, with  $t_i \in US$ , are most congested, and let any of those pins  $t_i \in US$ , for which  $\ell_0(t_i)$  pass over this interval, be chosen arbitrarily as a member of CS and be deleted from US.

At the first stage, the algorithm seeks wire routes composed of segments of level up to 1, as follows: Pick up any pin  $t_u \in US$ , and search for the following segments

- 1° a segment  $\ell_0(t_u)$  of level 0 which intersects any vertical wire segment already settled in the connected part of the net,
- 2° a segment  $\ell_0(t_u)$  of level 0 which meets a segment  $\ell_0(t_c)$  of level 0 for any  $t_c \in CS$ ,
- 3° a segment  $\ell_1(t_u)$  of level 1 which intersects a segment  $\ell_0(t_c)$  of level 0 for any  $t_c \in CS$ .

If such segments are found, then wire routes to connect  $t_u$  to the connected part can be defined, based on these segments. Choose the shortest\* of these routes, and add it newly to the connected part, with  $t_u$  transferred from US to CS. Repeat this process for the remaining pins in US.

At the next stage, wire routes composed of segments of level up to 2 are generated, as follows: Pick up any  $t_u \in US$ , and seek the following segments

- 1° a segment  $\ell_1(t_u)$  of level 1 which intersects any horizontal wire segment, already settled

\*. By the shortest route we mean the one which occupies the least number of cells.

- 2° so far, in the connected part of the net,
- 2° a segment  $\ell_2(t_u)$  of level 2 which intersects

a segment  $\ell_1(t_c)$  of level 1 for any  $t_c \in CS$ . If such segments are found, wire routes to connect  $t_u$  to the connected part can be defined on these segments. Choose the shortest of these routes, and add it newly to the connected part, with  $t_u$  transferred from US to CS. Repeat this for the remaining pins in US.

In the computer implementation of this algorithm, the process of finding segments of level 0, 1, and 2, which is dominant over the others, is conducted by making much use of the bit map, and every time any new wire route for a net is generated, all informations about its wire segments are stored in the core, one after another, which are transferred to the disk at the termination of the routing process for this net.

This algorithm realizes ordinarily 80-90 percent of the total wiring, and the rest is processed by another algorithm for the maze-running method, to be outlined below.

## B. Maze-Running Algorithm

The maze-running method, first proposed by Lee [10] and later modified by Geyer [11] and Rubin [12], consists essentially in finding a wire route between two cells by means of expansion around one until the other is reached, and hence does not fail to find a wire route to connect them whenever any one exists, although it requires processing time much more than the line-search method.

Any of the algorithms so far proposed aims mainly at seeking the shortest route and/or reducing the number of vias, and hence in its application to two-layer wiring boards, there may possibly occur a case when a fairly long vertical (horizontal) wire segment is drawn on the first (second) side\*\*, and this vertical (horizontal) segment prevents a certain number of horizontal (vertical) wire segments from being generated across it on the first (second) side later on, which lowers the wiring density of the board.

Considering this, the maze-running algorithm adopted in this system is divided into two phases:

The first phase is provided to avoid such a situation stated above, by introducing into it directional limitation facilities such that the vertical (horizontal) portion of any wire route between two consecutive vias, does not exceed totally some specified length on the first (second) side. Although by this limitation on generating wire pattern, a wire route, otherwise possible, might fail to be found, it should be claimed that in compensation for losing such a route, we can expect to get more routability by this, which is to avoid in advance making unnecessary obstacles for a number of wire routes to be generated later.

The second phase is mainly based on Chiba-Shirakawa-Ozaki [13], and is sophisticated such that its application to an incomplete net yields a tree-shaped wire pattern, which connects an isolated pin to the other isolated ones and the connected part of the net, whenever such a tree-shaped wire pattern exists. The following is a brief description of the second phase.

- 0°: Given a signal net  $T = \{t_1, t_2, \dots, t_n\}$  for which US and CS indicate the unconnected set and the connected set, respectively. Select a pin  $t_i \in US$ , let the cells on both sides occupied by this  $t_i$  be the start cells and the cells occupied by the other pins in US or by the connected part for this net the target cells. Mark the start cells and let them be the front cells.

- 1°: Repeat the following for each front cell while there exists one, and then go to 4°:

For any front cell  $(x, y)$ , examine whether or

\*\* Note that in this paper the first (second) side is principally for horizontal (vertical) wire segments.

not there is any target cell in its adjacent cells still unmarked, and let  $(x,y)$  be a non-front cell. If a target cell is found, then go to 2°, else mark all unoccupied cells adjacent to  $(x,y)$ , and add them to the front cells.

2°: If such a target cell is the one occupied by a pin in US, then mark it and add it to the front cells. Otherwise (in this case, such a target cell is occupied by the connected part), mark all the cells on the connected part, and add them to the front cells. Then go to 3°.

3°: If all the cells occupied by the pins in T are marked, then go to 4°, else return to 1°.

4°: From each marked target cell, settle the wire route backward to the start cells by tracing marked cells, one by one, in the reverse order of the search, and then halt.

At this stage, all the pins with their corresponding cells marked are proved to be connected to the start pin  $t_1$ .

In the computer implementation of this algorithm, the wiring area of a given board is paged into blocks of 32 rows by 32 columns, so that the blocks can be processed, page by page, in the core of the minicomputer in such a fashion that once the front line of the search enters into a different page to some depth from the boundary, that page is substituted for the old one in the core so as to be searched next.

## 7. Ordering Scheme

It is possible for a user to input his digitized or displayed manual layout into the system, instead of automatic wiring. Thus, the signal nets to be ordered are those which remain still unconnected after such a preassignment.

In this system the input data for such the remaining signal nets are divided into two classes: The first is DATA-1 consisting of those nets, given top priority by the user, which are to be arranged in the order of his instructions. The second is DATA-2, composed of the rest, which are left entirely for the ordering scheme.

Traditional ordering schemes tend to attach much importance to the length of potential wiring for each net. Apart from them, the ordering in this system esteems the relative relationship among the pin locations for each net.

The priority order for the nets depends on the assumption that ordinary DIP IC's are placed with dual pin-lines oriented vertically\*, and is given roughly as follows;

- I: nets within a certain horizontal spread,
- II: nets within a certain vertical spread,
- III: nets of high density pin-distribution.

A brief description of the ordering scheme is given as follows:

0°: Let U be a set of the nets in DATA-2, and for each net  $T_i = \{t_{i1}, t_{i2}, \dots, t_{in}\} \in U$ , with each pin  $t_{ik}$  placed at cell  $(x_{ik}, y_{ik})$ , let

$$M_x^i \triangleq \max_k \{x_{ik}\}, \quad m_x^i \triangleq \min_k \{x_{ik}\},$$

$$M_y^i \triangleq \max_k \{y_{ik}\}, \quad m_y^i \triangleq \min_k \{y_{ik}\},$$

$$D_x^i \triangleq M_x^i - m_x^i, \quad D_y^i \triangleq M_y^i - m_y^i.$$

Then, for specified constants  $a=20$ ,  $b=12$ , and  $c=50$ , sort U so as to be put into a list  $E = (E_1, E_2, E_3, E_4)$  by the following procedure.

1°: For  $S_1 \triangleq \{T_i \in U \mid D_x^i \leq a\}$ , make a list  $E_1$  in ascend-

ing order of  $D_y^i$ , and put  $U \leftarrow U - S_1$ .

2°: For  $S_2 \triangleq \{T_i \in U \mid D_y^i \leq b\}$ , make a list  $E_2$  in ascending order of  $D_x^i$ , and put  $U \leftarrow U - S_2$ .

3°: For  $S_3 \triangleq \{T_i \in U \mid D_x^i \leq c\}$ , make a list  $E_3$  in ascending order of  $D_y^i$ , and put  $U \leftarrow U - S_3$ .

4°: For each  $T_i \in U$ , calculate  $d_i \triangleq (D_x^i + D_y^i)/n$ , where  $n \triangleq |T_i|$ , and make a list  $E_4$  in ascending order of  $d_i$ . Here, it should be added that the list E thus obtained is permitted to be altered in part by the user.

## 8. Interactive Editing Routine

Interactive editing routine provides two kinds of facilities for graphics updating; the first is S-GRAPH to update the current wire pattern symbolically displayed on the screen, based on the design data base, and the second is C-GRAPH to complete the final composite artwork pattern to be inputted to a photo-plotter, based on Artwork File.

In the process of implementing S-GRAPH, once a user attains 100 percent wiring, he has only to modify the current wire pattern until he is satisfied with it, or until he is convinced that it may be free from any trouble latently possible in the subsequent stage. On the other hand, while 100 percent wiring is not still realized, he may be devoted to finding another better pattern by adding new wire segments and/or modifying the current pattern. However, especially when the wiring density is high, such a manual layout, which is a prevailing policy, is so complicated that it is fairly time-consuming, and furthermore, rather liable to wiring errors. To cope with this difficulty, this routine is organized so as to supply for the user a variety of sophisticated instructions convenient for modifying wire patterns, by which he can easily make an open space in the current wire pattern in order that a wire route still missing in it can later be laid out automatically by applying either the line-search or maze-running routine.

The instructions prepared in this system for graphics updating are listed as shown in Table 2, and are classified as

—Window	—Select	—Add
—Enlarge	—Move	—Grid-on
—Reduce	—Delete	—Scale
—Shift	—Repeat.	

Table 2 Operating Command List for Graphics Updating

### 1. Control Mode

XEQ	Execute buffered commands
MKDR	Load drawing from master file to working drawing
STOR	Store working drawing to back-up file
EDIT	Enter edit mode
EXIT	Close edit mode

### 2. Edit Mode

#### A. Edit Control Mode

XEQE	Execute buffered commands and redraw picture
DBUF	Delete all buffered commands
CTRL	Enter control mode

#### B. Window Manipulation

FIT	Fit drawing into screen
WIND	Enlarge specified part of drawing into window
DWIN	Decrease magnitude by half
SHFU	Move window up by 1/2 window height
SHFD	Move window down by 1/2 window height
SHFL	Move window left by 1/2 window width
SHFR	Move window right by 1/2 window width

\* In this case, there are far more obstacles for horizontal wire segments than vertical ones, in general.



### C. Level Manipulation

EDTL	Make specified levels editable
XEDT	Exclude editable levels from display

### D. Select Manipulation

SELA	Select all components
SELC	Select center of specified component
SELE	Select end of specified component
SELP	Select specified pins or through-holes
SNET	Select all components of specified signal net
UNSA	Unselect all selected components
UNSC	Unselect all selected centers of specified components
UNSE	Unselect all selected ends of specified components
UNSP	Unselect specified pins or through-holes selected

### E. Component Manipulation

ADD	Add component on specified point
DELT	Delete selected components
MOVE	Move selected components
SWAP	Change existing level of selected components
CHGS	Change signal net selected
CHGD	Change spindle size for selected pins
CHGR	Change photo aperture for selected pins
CHGL	Change photo aperture for selected lines
NAMS	Display statistics for selected component
LOCK	Switch to prohibit any oblique line
CUTL	Cut specified line into two lines
DNET	Display all components of selected signal net
DTBL	Display details of used parts

### F. Grids and Rulers Manipulation

SCAL	Display x, y axes
GRID	Grid display switch
GDON	Grid on-off switch
DLOC	Display abs. coordinates at specified point

User interaction with the editing commands is carried out through the use of a stylus. These available instructions are displayed at the right of the screen, and the user can select one of them with the stylus. At the top of the screen, accepted instructions are also displayed.

Some implementation examples are shown in Figs. 3-5.

In the computer implementation, a magnetic disk data segmentation facility is employed, whenever necessary, which enables only the required portions of the whole wire pattern to be in core while the rest is kept on backing store.

C-GRAPH is a routine to complete an artwork layout pattern of a board at the final stage, based on Artwork File. In its implementation, the layout pattern is realistically displayed, such that each wire segment has a specified width and each plated-through hole or via is in the shape of a circle with a specified diameter. To manipulate the layout associated with unstandardized circuit modules, facilities are available to alter the grid size, at the desire of a user, with precision up to 0.01 mm.

In addition to the commands listed in Table 2, the following ones are also available in this C-GRAPH, which are properly provided;

DRUL	Display ruler between two specified points
WGRD	Change working grid
REAL	Switch display mode from (to) symbolic to (from) realistic (or composite)
DIVD	Divide the straight line between two specified points into k equally parts, so that k line segments with the same width can pass across the line

Some implemented examples are shown in Figs. 6-8.

### 9. Replacement Routine

This is optionally provided for modifying the placement given by a user, and is to be applied mainly in a case when the wiring performance through the first run of the automatic routing scheme is too bad, such as 80 percent or less.

If the user wants to replace modules on the board, he can accomplish it by the use of graphics. In this routine, a display of the modules on the board can be viewed. In order to visually analyze the placement quality, minimum spanning trees for some or all of the nets can be displayed. Through the use of interactive display techniques, the effects of module movement and exchange can be instantaneously realized.

How to improve the placement is illustrated in the following with the use of an actual example. Fig. 9 shows the displayed placement of a given digitized input, in which each straight line connecting two pins corresponds to a connection requirement, and the profiles at the right and the bottom depict the congestions of these straight lines crossing corresponding rows and columns, respectively. For this placement, a sequence of processes of exchanging a pair of IC's and/or moving an IC to an idle position are conducted so that the total length of the shortest trees (minimum spanning trees) for all nets can be reduced, where a tree for a net is not a Steiner's one, but simply a spanning one in the complete graph which is defined by all pins in the net as its vertices and edges between all pairs of such pins, each with a weight of its distance. The total lengths of the shortest trees in the last placement and in the present one are displayed at the upper right-hand corner. Fig. 10 shows a replacement result for Fig. 9, in which the list of IC's displayed at the right indicates those which are newly selected for movement from the last placement, and the total length of the shortest trees is seen to be reduced from the initial 40,364 to 29,940.

### 10. Implemented Results

This system has been at work, and more than 300 two-layer boards of manufacturing use have been laid out by this, almost all with 100 percent routing realized, where all those boards without 100 percent realization are for equipments to be made on an experimental basis, instead of finished products, for which 100 percent routing is not always sought with persistence.

To see the wirability of the router in this system, we have compared it with APD-W of IBM, Japan, which is a program system of batch processing and is most extensively used in Japan. Statistics of all the five sample boards are as follows:

Size: 200 rows by 204 columns

No. of modules mounted:

# connectors	22-pin×4	} equivalently 81 14-pin IC's
# IC's	16-pin×53	
# LSI's	24-pin×8	

Module density: 1.26 inch<sup>2</sup>/14-pin IC

Implemented results for all these five boards are shown in Table 3, where #connections indicates the total sum of required connection pairs, that is, the total sum of  $|T_i| - 1$  for all nets  $T_i$ , LINE and MAZE denote wiring performances of line-search and maze-running portions, respectively (associated with this system these two routines are without interaction).

Table 4 shows a part of the implemented results by this system, where Wiring Area indicates the effective area for the routing process, Module Density denotes area of inch<sup>2</sup>/IC equivalent, and Rerun of MAZE means the wiring portion performed by the maze-running routine after modifications on wire pattern in S-GRAPH.

Figs. 11 and 12 show the artwork patterns on the first and second sides, respectively, of a board outputted by a photo-plotter.



Table 3 Comparison of this system with APD-W

Datum No.	1		2		3		4		5	
Systems	APD-W	This	APD-W	This	APD-W	This	APD-W	This	APD-W	This
#nets	157		179		249		202		178	
#connections	324		352		420		343		370	
#used-pins/IC	7.9		8.6		10.3		8.7		8.8	
LINE	79.63%	94.75%	83.81%	98.30%	75.71%	90.21%	78.43%	93.40%	79.19%	90.54%
MAZE	16.36	2.78	13.92	1.42	16.43	4.33	16.61	3.70	15.40	5.95
Total	95.99	97.53	97.73	99.72	92.14	94.54	95.04	97.10	94.59	96.49
#isolated-pins	15	9	11	1	35	32	21	15	24	21

(Note here that all these boards have been completed with 100 percent routing by the next run of MAZE in S-GGRAPH.)

Table 4 Implemented Results

Board No.	Board Size (inch)	Wiring Area (inch <sup>2</sup> )	#pins	Module Density (inch <sup>2</sup> /IC)	#nets	#connections	Wiring Performance (%)			
							LINE	1st MAZE	Rerun of MAZE	TOTAL
1	9.5 × 9.1	65.4	1092	0.84	137	190	82.11	15.78	2.11	100
2	12.3 × 11.6	130.9	2283	0.80	220	338	74.26	17.46	8.28	100
3	6.1 × 11.4	57.1	1306	0.61	190	335	76.12	18.51	5.37	100
4	9.5 × 10.9	66.9	1196	0.78	123	169	78.70	15.38	5.92	100
5	9.5 × 10.9	98.9	1487	0.93	106	169	78.11	14.79	7.10	100
6	10.3 × 9.0	83.7	1160	1.01	294	530	91.32	8.68	0	100
7	9.5 × 11.0	43.2	456	1.33	68	101	97.03	2.97	0	100
8	12.3 × 12.8	141.6	1768	1.12	260	507	97.44	2.56	0	100
9	12.4 × 11.7	119.9	1496	1.12	237	407	73.71	17.69	8.60	100
10	12.3 × 11.6	112.5	1665	0.95	319	652	90.49	6.90	2.61	100
11	12.5 × 12.5	119.8	1811	0.93	287	517	74.85	18.38	6.38	99.61
12	8.5 × 12.9	93.7	1105	1.19	218	316	87.66	8.86	3.48	100
13	9.5 × 11.5	82.8	1457	0.80	316	604	78.97	16.23	4.80	100
14	12.4 × 12.0	124.1	2270	0.77	263	467	83.08	8.57	7.75	99.36
15	12.4 × 11.6	128.1	1617	1.11	345	635	78.74	14.96	6.30	100

### 11. Conclusion

We have proposed a minicomputerized automatic layout system for two-layer printed wiring boards, and outlined its functional organization. This system works in practice, and almost all the wiring boards of manufacturing use so far processed by this system have been completed with 100 percent routing. Thus, it is confirmed that this system is contributing much toward reducing cost and time incurred in laying out wire patterns.

Development is continuing on more sophisticated automatic routing and placement algorithms, as well as an efficient scheme for assignment of gates to circuit modules, to be optionally included in this system.

### Acknowledgement

The authors would like to express their appreciation to C. Suzuki and S. Misaka, Sharp Co., for continuous encouragement and valuable suggestion, and they also wish to thank H. Kubo and S. Yamamoto, Sharp Co., for their invaluable assistance in making programs in Interactive Editing Routine, Maze-Running Routine, and Replacement Routine.

### References

- [1] M.A.Brreuer (ed.), *Design Automation of Digital*

Systems, vol. 1, Englewood Cliffs, N. J.: Prentice-Hall, 1972.

- [2] H.Nakahara, "Computer-aided interconnection routing: General survey of the stage-of-the-art," *Networks*, vol. 2, pp. 167-183, 1972.
- [3] D.W.Hightower, "The interconnection problem—A tutorial," *Proc. Design Automation Workshop*, pp.1-12, 1973.
- [4] L.Abel, "On the automated layout of multi-layer planar wiring and a related graph coloring problem," *Coordinated Sci. Lab. Rept.*, No. R-546, Univ. Ill., 1972.
- [5] B.R.Rau, "A new philosophy for interconnection on multilayer boards," *Proc. Design Automation Conf.*, pp. 225-231, 1976.
- [6] J.R.Allen, "A topological adaptable cellular router," *Ibid*, pp. 161-167, 1976.
- [7] G.L.Patterson and B.H.Phillips, "A proven operational CAD system for p.w.b. design—based on a minicomputer and featuring fully automatic placement and routing," *Ibid*, pp. 259-264, 1976.
- [8] K.Mikami and K.Tabuchi, "A computer program for optimal routing of printed circuit conductors," *Proc. IFIP Congress*, pp. 1475-1478, 1968.
- [9] H.Yamamura, I.Shirakawa, and H.Ozaki, "A line-search method for the route connecting problem on 2-layer printed circuit boards," *IECE Trans.*, vol. 57-A,