NASA/TM-2000-209036

User's Guide for Computer Program That Routes Signal Traces

David R. Hedgley, Jr.
NASA Dryden Flight Research Center
Edwards, California

December 2000

The NASA STI Program Office…in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the lead center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and mission, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results…even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at ***http://www.sti.nasa.gov***

- E-mail your question via the Internet to help@sti.nasa.gov

- Fax your question to the NASA Access Help Desk at (301) 621-0134

- Telephone the NASA Access Help Desk at (301) 621-0390

- Write to:
  NASA Access Help Desk
  NASA Center for AeroSpace Information
  7121 Standard Drive
  Hanover, MD 21076-1320

NASA/TM-2000-209036

User's Guide for Computer Program That Routes Signal Traces

David R. Hedgley, Jr.
NASA Dryden Flight Research Center
Edwards, California

National Aeronautics and
Space Administration

Dryden Flight Research Center
Edwards, California 93523-0273

December 2000

Available from the following:

NASA Center for AeroSpace Information (CASI)
7121 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service (NTIS)
5285 Port Royal Road
Springfield, VA 22161-2171
(703) 487-4650

ABSTRACT

This disk contains both a FORTRAN computer program and the corresponding user's guide that facilitates both its incorporation into your system and its utility. The computer program represents an efficient algorithm that routes signal traces on layers of a printed circuit with both through-pins and surface mounts. The computer program included is an implementation of the ideas presented in the theoretical paper titled A Formal Algorithm for Routing Signal Traces on a Printed Circuit Board, NASA TP-3639 published in 1996. The computer program in the "connects" file can be read with a FORTRAN compiler and readily integrated into software unique to each particular environment where it might be used.

README:

This file (readme) addresses the functions of all the files on this disk. The files are listed below with a description of each one.

(1) The "readme" file describes other files on the disk and contains the abstract.

(2) The "uguide" file describes how to implement the router software.

(3) The "testpgm" file is a source file that contains a sample calling program and the router software (which is a composite of subroutines).

(4) The "connects" file contains the input data for the calling program that is noted in the testpgm file.

(5) The "sample" file contains the output with which to validate that correct installation of the software has been accomplished. (This file should be saved before the testpgm file is executed.)

(6) A pdf of NASA Technical Paper 3639, A Formal Algorithm for Routing Traces on a Printed Circuit Board, published in 1996, with a routing map of signal traces as the appendix of that report.

To make these files available to the largest number of computer users, each file appears in two formats, as a pdf file and as a Microsoft Word text file. The NASA TP-3639 appears only as a pdf file.

Note: Adobe Acrobat Reader, required for viewing pdf files, can be downloaded free of charge at www.adobe.com.

419	3.350000	2.300000	0.220000					
0.5000	0.2000	0.5000	0.3000	0.5000	0.4000	0.5000	0.5000	0.5000
0.6000	0.5000	0.7000	0.5000	0.8000	0.7000	0.2000	0.7000	0.3000
0.7000	0.4000	0.7000	0.5000	0.7000	0.6000	0.7000	0.7000	0.7000
0.8000	0.5000	1.0500	0.5000	1.1500	0.5000	1.2500	0.5000	1.3500
0.5000	1.4500	0.5000	1.5500	0.5000	1.6500	0.7000	1.0500	0.7000
1.1500	0.7000	1.2500	0.7000	1.3500	0.7000	1.4500	0.7000	1.5500
0.7000	1.6500	0.5000	1.9000	0.5000	2.0000	0.5000	2.1000	0.5000
2.2000	0.5000	2.3000	0.5000	2.4000	0.5000	2.5000	0.7000	1.9000
0.7000	2.0000	0.7000	2.1000	0.7000	2.2000	0.7000	2.3000	0.7000
2.4000	0.7000	2.5000	0.8500	0.2000	0.8500	0.3000	0.8500	0.4000
0.8500	0.5000	0.8500	0.6000	0.8500	0.7000	0.8500	0.8000	1.0500
0.2000	1.0500	0.3000	1.0500	0.4000	1.0500	0.5000	1.0500	0.6000
1.0500	0.7000	1.0500	0.8000	0.8500	1.0500	0.8500	1.1500	0.8500
1.2500	0.8500	1.3500	0.8500	1.4500	0.8500	1.5500	0.8500	1.6500
1.0500	1.0500	1.0500	1.1500	1.0500	1.2500	1.0500	1.3500	1.0500
1.4500	1.0500	1.5500	1.0500	1.6500	0.8500	1.9000	0.8500	2.0000
0.8500	2.1000	0.8500	2.2000	0.8500	2.3000	0.8500	2.4000	0.8500
2.5000	1.0500	1.9000	1.0500	2.0000	1.0500	2.1000	1.0500	2.2000
1.0500	2.3000	1.0500	2.4000	1.0500	2.5000	1.2000	0.2000	1.2000
0.3000	1.2000	0.4000	1.2000	0.5000	1.2000	0.6000	1.2000	0.7000
1.2000	0.8000	1.4000	0.2000	1.4000	0.3000	1.4000	0.4000	1.4000
0.5000	1.4000	0.6000	1.4000	0.7000	1.4000	0.8000	1.2000	1.0500
1.2000	1.1500	1.2000	1.2500	1.2000	1.3500	1.2000	1.4500	1.2000
1.5500	1.2000	1.6500	1.4000	1.0500	1.4000	1.1500	1.4000	1.2500
1.4000	1.3500	1.4000	1.4500	1.4000	1.5500	1.4000	1.6500	1.2000
1.9000	1.2000	2.0000	1.2000	2.1000	1.2000	2.2000	1.2000	2.3000
1.2000	2.4000	1.2000	2.5000	1.4000	1.9000	1.4000	2.0000	1.4000
2.1000	1.4000	2.2000	1.4000	2.3000	1.4000	2.4000	1.4000	2.5000
1.5500	0.2000	1.5500	0.3000	1.5500	0.4000	1.5500	0.5000	1.5500
0.6000	1.5500	0.7000	1.5500	0.8000	1.7500	0.2000	1.7500	0.3000
1.7500	0.4000	1.7500	0.5000	1.7500	0.6000	1.7500	0.7000	1.7500
0.8000	1.5500	1.0500	1.5500	1.1500	1.5500	1.2500	1.5500	1.3500
1.5500	1.4500	1.5500	1.5500	1.5500	1.6500	1.7500	1.0500	1.7500
1.1500	1.7500	1.2500	1.7500	1.3500	1.7500	1.4500	1.7500	1.5500
1.7500	1.6500	1.5500	1.9000	1.5500	2.0000	1.5500	2.1000	1.5500
2.2000	1.5500	2.3000	1.5500	2.4000	1.5500	2.5000	1.7500	1.9000
1.7500	2.0000	1.7500	2.1000	1.7500	2.2000	1.7500	2.3000	1.7500
2.4000	1.7500	2.5000	1.9000	0.2000	1.9000	0.3000	1.9000	0.4000
1.9000	0.5000	1.9000	0.6000	1.9000	0.7000	1.9000	0.8000	2.1000
0.2000	2.1000	0.3000	2.1000	0.4000	2.1000	0.5000	2.1000	0.6000
2.1000	0.7000	2.1000	0.8000	1.9000	1.0500	1.9000	1.1500	1.9000
1.2500	1.9000	1.3500	1.9000	1.4500	1.9000	1.5500	1.9000	1.6500
2.1000	1.0500	2.1000	1.1500	2.1000	1.2500	2.1000	1.3500	2.1000
1.4500	2.1000	1.5500	2.1000	1.6500	1.9000	1.9000	1.9000	2.0000
1.9000	2.1000	1.9000	2.2000	1.9000	2.3000	1.9000	2.4000	1.9000
2.5000	2.1000	1.9000	2.1000	2.0000	2.1000	2.1000	2.1000	2.2000
2.1000	2.3000	2.1000	2.4000	2.1000	2.5000	2.2500	0.2000	2.2500
0.3000	2.2500	0.4000	2.2500	0.5000	2.2500	0.6000	2.2500	0.7000
2.2500	0.8000	2.4500	0.2000	2.4500	0.3000	2.4500	0.4000	2.4500
0.5000	2.4500	0.6000	2.4500	0.7000	2.4500	0.8000	2.2500	1.0500
2.2500	1.1500	2.2500	1.2500	2.2500	1.3500	2.2500	1.4500	2.2500
1.5500	2.2500	1.6500	2.4500	1.0500	2.4500	1.1500	2.4500	1.2500
2.4500	1.3500	2.4500	1.4500	2.4500	1.5500	2.4500	1.6500	2.2500
1.9000	2.2500	2.0000	2.2500	2.1000	2.2500	2.2000	2.2500	2.3000
2.2500	2.4000	2.2500	2.5000	2.4500	1.9000	2.4500	2.0000	2.4500
2.1000	2.4500	2.2000	2.4500	2.3000	2.4500	2.4000	2.4500	2.5000

2.6000	0.2000	2.6000	0.3000	2.6000	0.4000	2.6000	0.5000	2.6000
0.6000	2.6000	0.7000	2.6000	0.8000	2.8000	0.2000	2.8000	0.3000
2.8000	0.4000	2.8000	0.5000	2.8000	0.6000	2.8000	0.7000	2.8000
0.8000	2.6000	1.0500	2.6000	1.1500	2.6000	1.2500	2.6000	1.3500
2.6000	1.4500	2.6000	1.5500	2.6000	1.6500	2.8000	1.0500	2.8000
1.1500	2.8000	1.2500	2.8000	1.3500	2.8000	1.4500	2.8000	1.5500
2.8000	1.6500	2.6000	1.9000	2.6000	2.0000	2.6000	2.1000	2.6000
2.2000	2.6000	2.3000	2.6000	2.4000	2.6000	2.5000	2.8000	1.9000
2.8000	2.0000	2.8000	2.1000	2.8000	2.2000	2.8000	2.3000	2.8000
2.4000	2.8000	2.5000	2.9500	0.2000	2.9500	0.3000	2.9500	0.4000
2.9500	0.5000	2.9500	0.6000	2.9500	0.7000	2.9500	0.8000	3.1500
0.2000	3.1500	0.3000	3.1500	0.4000	3.1500	0.5000	3.1500	0.6000
3.1500	0.7000	3.1500	0.8000	2.9500	1.0500	2.9500	1.1500	2.9500
1.2500	2.9500	1.3500	2.9500	1.4500	2.9500	1.5500	2.9500	1.6500
3.1500	1.0500	3.1500	1.1500	3.1500	1.2500	3.1500	1.3500	3.1500
1.4500	3.1500	1.5500	3.1500	1.6500	2.9500	1.9000	2.9500	2.0000
2.9500	2.1000	2.9500	2.2000	2.9500	2.3000	2.9500	2.4000	2.9500
2.5000	3.1500	1.9000	3.1500	2.0000	3.1500	2.1000	3.1500	2.2000
3.1500	2.3000	3.1500	2.4000	3.1500	2.5000	3.3000	0.2000	3.3000
0.3000	3.3000	0.4000	3.3000	0.5000	3.3000	0.6000	3.3000	0.7000
3.3000	0.8000	3.5000	0.2000	3.5000	0.3000	3.5000	0.4000	3.5000
0.5000	3.5000	0.6000	3.5000	0.7000	3.5000	0.8000	3.3000	1.0500
3.3000	1.1500	3.3000	1.2500	3.3000	1.3500	3.3000	1.4500	3.3000
1.5500	3.3000	1.6500	3.5000	1.0500	3.5000	1.1500	3.5000	1.2500
3.5000	1.3500	3.5000	1.4500	3.5000	1.5500	3.5000	1.6500	3.3000
1.9000	3.3000	2.0000	3.3000	2.1000	3.3000	2.2000	3.3000	2.3000
3.3000	2.4000	3.3000	2.5000	3.5000	1.9000	3.5000	2.0000	3.5000
2.1000	3.5000	2.2000	3.5000	2.3000	3.5000	2.4000	3.5000	2.5000
3.6500	0.2000	3.6500	0.3000	3.6500	0.4000	3.6500	0.5000	3.6500
0.6000	3.6500	0.7000	3.6500	0.8000	3.8500	0.2000	3.8500	0.3000
3.8500	0.4000	3.8500	0.5000	3.8500	0.6000	3.8500	0.7000	3.8500
0.8000	3.6500	1.0500	3.6500	1.1500	3.6500	1.2500	3.6500	1.3500
3.6500	1.4500	3.6500	1.5500	3.6500	1.6500	3.8500	1.0500	3.8500
1.2500	3.8500	1.3500	3.8500	1.4500	3.8500	1.5500	3.8500	1.6500
3.6500	1.9000	3.6500	2.0000	3.6500	2.1000	3.6500	2.2000	3.6500
2.3000	3.6500	2.4000	3.6500	2.5000	3.8500	1.9000	3.8500	2.0000
3.8500	2.1000	3.8500	2.2000	3.8500	2.3000	3.8500	2.4000	3.8500
2.5000								
0.5000	1.6500	2.1000	0.3000	0.5000	1.9000	1.5500	0.5000	0.5000
2.3000	0.5000	2.3000	1.5500	0.8000	2.8000	0.2000	0.7000	2.3000
3.3000	0.4000	0.7000	1.5500	0.7000	1.9000	2.9500	0.7000	2.8000
0.8000	3.1500	1.0500	3.3000	1.1500	2.1000	1.2500	3.1500	1.3500
2.1000	1.4500	2.8000	1.5500	1.7500	1.6500	0.7000	2.4000	2.1000
1.1500	1.9000	1.2500	1.9000	1.3500	3.6500	1.4500	3.1500	1.5500
1.7500	1.6500	3.6500	1.9000	0.5000	0.3000	2.6000	2.1000	0.5000
0.4000	1.5500	2.3000	0.5000	0.7000	0.5000	1.2500	2.4500	1.9000
0.7000	0.4000	0.7000	0.4000	2.2500	2.2000	0.7000	0.6000	0.7000
0.7000	0.7000	1.3500	0.8500	1.2500	2.4500	0.3000	0.8500	2.0000
2.8000	0.5000	0.8500	2.1000	0.8500	1.9000	2.8000	0.8000	2.4500
0.2000	1.0500	1.4500	1.0500	2.1000	2.6000	0.5000	2.6000	0.6000
3.1500	0.7000	2.8000	0.8000	0.8500	2.2000	3.8500	1.1500	0.8500
2.4000	0.8500	0.3000	0.8500	0.4000	2.2500	1.5500	2.2500	1.6500
1.0500	2.4000	3.8500	1.1500	1.0500	2.4000	3.6500	1.3500	1.0500
2.5000	2.4500	1.5500	1.0500	0.5000	0.8500	0.2000	0.8500	0.5000
1.9000	2.1000	0.8500	0.6000	0.8500	0.3000	2.4500	2.4000	0.8500
1.4500	1.0500	0.6000	1.0500	0.7000	3.6500	2.1000	3.1500	2.2000
2.4500	2.3000	2.2500	2.4000	3.6500	2.5000	2.4500	0.2000	1.2000

2.4000	3.3000	0.4000	2.8000	0.5000	1.2000	2.0000	3.3000	0.7000
3.5000	0.8000	3.5000	0.2000	2.6000	2.0000	3.3000	0.8000	3.5000
2.2000	3.3000	2.2000	2.8000	0.4000	3.8500	0.7000	0.8500	0.3000
3.5000	1.9000	2.6000	1.3500	2.1000	1.5500	2.6000	0.2000	3.6500
2.0000	2.8000	0.2000	3.6500	2.3000	2.9500	2.2000	2.6000	1.5500
3.3000	2.2000	3.3000	0.3000	3.3000	1.2500	2.2500	0.2000	3.1500
2.5000	0.7000	0.2000	3.8500	1.1500	3.6500	2.1000	3.8500	2.0000
2.2500	2.1000	2.4500	0.2000	3.8500	0.5000	2.1000	1.2500	2.2500
1.4500	3.3000	1.0500	3.5000	1.5500	3.1500	1.3500	3.3000	1.1500
1.2000	1.0500	2.9500	1.0500	3.1500	1.2500	2.6000	0.3000	2.2500
2.5000	2.1000	1.2500	2.4500	1.4500	2.6000	0.8000	1.0500	2.1000
1.4000	2.0000	2.8000	0.3000	1.5500	1.5500	0.5000	0.5000	3.1500
1.9000	0.5000	2.5000	0.8500	0.4000	2.6000	0.5000	2.6000	2.0000
0.8500	0.8000	3.1500	0.5000	3.5000	2.5000	2.8000	0.7000	0.7000
2.0000	0.7000	1.0500	2.8000	0.5000	1.0500	0.7000	2.9500	0.7000
0.8500	1.9000	2.6000	2.2000	1.9000	1.2500	3.8500	2.5000	2.1000
1.4500	2.4500	1.5500	1.0500	1.5500	1.9000	0.6000	0.7000	2.0000
3.3000	2.5000	2.1000	0.7000	3.1500	1.1500	2.8000	1.4500	3.8500
1.2500	2.6000	1.3500	3.6500	2.4000	3.1500	0.8000	3.3000	2.0000
1.4000	1.6500	3.5000	1.6500	2.4500	1.3500	2.8000	0.3000	0.7000
1.3500	2.4500	1.9000	1.7500	1.3500	0.8500	0.7000	2.8000	2.1000
1.4000	1.0500	3.8500	0.5000	1.2000	2.2000	2.9500	1.5500	2.9500
0.6000	2.9500	1.9000	2.2500	2.4000	2.4500	2.3000	1.2000	1.1500
1.4000	0.6000	1.5500	0.3000	3.1500	1.4500	1.9000	2.2000	3.8500
1.1500	3.8500	1.3500	3.3000	1.0500	1.5500	0.5000	3.1500	1.9000
0.7000	1.6500	2.9500	0.5000	1.0500	1.5500	3.5000	0.2000	2.4500
1.3500	0.7000	2.4000	0.7000	2.1000	1.4000	1.3500	2.2500	0.6000
2.6000	1.4500	3.3000	0.6000	3.5000	0.5000	3.6500	2.4000	3.8500
2.2000	3.1500	2.0000	2.4500	2.2000	3.1500	2.5000	3.5000	2.3000
3.6500	1.2500	1.4000	2.4000	3.6500	2.0000	3.6500	1.9000	1.4000
0.7000	3.6500	1.2500	3.6500	1.6500	1.7500	0.3000	2.6000	1.9000
0.7000	0.2000	1.9000	2.4000	1.5500	1.6500	2.4500	0.5000	3.8500
1.4500	3.1500	1.2500	1.0500	1.1500	0.5000	2.4000	0.8500	2.1000
2.1000	0.5000	1.2000	1.5500	2.9500	2.5000	0.5000	1.2500	1.0500
2.5000	3.5000	2.5000	3.3000	1.2500	3.3000	1.5500	3.6500	1.6500
3.8500	1.6500	3.3000	2.0000	2.9500	1.2500	3.3000	0.5000	1.0500
2.4000	3.1500	0.6000	1.5500	2.5000	0.5000	0.5000	3.3000	0.6000
2.1000	1.9000	2.2500	1.1500	1.7500	0.2000	3.8500	1.6500	2.8000
2.0000	3.1500	2.3000	1.0500	1.1500	2.1000	0.7000	1.7500	0.8000
3.1500	1.1500	2.9500	1.3500	3.5000	1.4500	0.5000	2.2000	3.3000
1.4500	1.9000	1.4500	1.5500	2.5000	0.8500	2.3000	1.9000	2.5000
3.3000	0.4000	1.0500	2.3000	3.1500	0.5000	1.0500	2.2000	0.7000
2.5000	0.8500	0.4000	1.5500	2.3000	2.1000	0.8000	2.2500	2.2000
2.4500	2.5000	3.8500	2.1000	2.8000	1.1500	1.7500	0.2000	1.2000
1.5500	3.6500	1.4500	0.7000	0.4000	0.7000	2.4000	3.1500	0.8000
3.5000	1.5500	0.7000	2.3000	3.8500	1.6500	2.1000	2.0000	3.5000
1.5500	0.8500	0.6000	2.8000	1.5500	1.7500	0.2000	3.6500	2.1000
1.7500	0.7000	3.6500	2.4000	1.7500	2.4000	0.8500	1.4500	2.2500
1.9000	2.1000	1.1500	2.4500	1.9000	2.4500	1.3500	2.4500	2.3000
2.2500	1.3500	1.5500	0.2000	2.8000	2.0000	2.1000	2.1000	1.9000
0.2000	3.3000	2.4000	1.7500	2.2000	1.2000	0.3000	1.4000	1.1500
1.4000	0.4000	2.6000	2.3000	1.7500	2.2000	1.7500	2.5000	2.2500
0.4000	2.2500	2.3000	2.8000	0.7000	0.7000	1.1500	0.5000	0.5000
2.8000	1.1500	3.3000	1.0500	3.6500	0.3000	0.7000	1.0500	1.4000
1.0500	2.9500	0.4000	1.9000	1.1500	1.7500	1.2500	2.2500	2.5000
2.4500	2.0000	3.6500	0.2000	1.4000	0.6000	2.8000	2.4000	0.7000
0.8000	1.9000	2.1000	1.9000	2.5000	0.5000	0.7000	1.5500	1.0500

3.1500	2.4000	0.5000	0.6000	2.9500	1.0500	3.1500	1.1500	2.1000
1.0500	2.6000	1.3500	1.0500	1.1500	2.2500	0.5000	2.6000	0.6000
2.6000	2.5000	3.5000	0.2000	0.5000	2.0000	2.1000	0.5000	1.7500
1.2500	3.5000	0.3000	0.7000	1.2500	3.1500	0.3000	1.4000	0.5000
1.2000	2.2000	2.1000	1.0500	2.2500	1.9000	2.8000	2.3000	2.1000
0.8000	2.6000	1.2500	2.1000	0.5000	2.9500	1.3500	2.2500	2.1000
3.8500	1.5500	1.4000	2.0000	1.5500	0.8000	1.2000	1.3500	2.9500
1.4500	2.1000	0.7000	1.2000	0.6000	1.0500	1.4500	3.8500	1.0500
1.5500	2.1000	3.8500	1.5500	2.4500	1.6500	2.2500	2.3000	1.2000
0.4000	1.2000	0.2000	3.8500	2.3000	2.2500	0.7000	1.9000	0.4000
0.7000	0.7000	0.5000	0.4000	2.4500	1.6500	1.9000	1.4500	2.2500
0.5000	3.1500	2.1000	1.4000	0.6000	2.4500	2.2000	0.8500	1.5500
1.5500	1.5500	1.9000	1.2500	1.5500	0.3000	3.6500	2.2000	1.4000
0.3000	1.9000	1.1500	2.9500	1.1500	1.0500	2.2000	0.7000	2.0000
2.1000	1.1500	1.9000	0.2000	1.5500	2.0000	1.9000	0.4000	0.7000
1.3500	0.8500	2.1000	2.1000	2.0000	2.8000	1.3500	2.6000	0.6000
0.5000	1.9000	1.9000	2.5000	2.2500	2.2000	3.3000	1.6500	1.9000
2.1000								
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000				
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000				

dx=.35 dy=2.3 percent=.22

AVERAGE NUMBER OF FEEDTHROUGHS PER TRACE = 1.1840

(ACTUAL DISTANCE/MANHATTAN DISTANCE) = 0.9620

PERCENTAGE OF COMPLETION = 0.2983

ACTUAL NUMBER OF COMPLETED TRACES = 125

TOTAL NUMBER OF ATTEMPTED TRACES = 419

```
c    This demonstration calling program expects an entrance of '0'
c    at the time the inquiry is made which gives statistics consistent
c    with the sample file.......DO NOT PLOT!!!!!
c    Once you change the close and setup routines to reflect your
c    environment, then you can plot on your device.

      DIMENSION MATRIX(24964),INDEX(3950),DX(419),DY(419),
     1XM(419),YM(419),TYFB(419),TYFE(419),WK(419),IWK1(419),
     2IWK2(419),IWK3(419)
      COMMON/DAVID/NCO,PDW,N1,XLL(20),YLL(20),XUR(20),YUR(20),BW,
     1BH,NOO,NL1,NL2,RESL
      COMMON/BEG/IBEG,IEND
      open(unit=7,file='sample')
      OPEN(UNIT=1,FILE='connects')
 2288 FORMAT(9F8.7)
      READ(1,2201)NCO,TXX,TYY,PER
      READ(1,2288)(XM(I),YM(I),I=1,NCO)
      READ(1,2288)(DX(I),DY(I),I=1,NCO)
      READ(1,2288)(TYFB(I),I=1,NCO)
      READ(1,2288)(TYFE(I),I=1,NCO)
      RESL=.05
      PDW=.04
      BH=TYY+.6
      BW=TXX+.6
      N1=0
      XLL(1)=0
      YLL(1)=0
      XUR(1)=0
      YUR(1)=0
      XLL(2)=0
      YLL(2)=0
      XUR(2)=0
      YUR(2)=0
      NOO=1
      NL1=1
      NL2=0
      BB=AMAX1(BH,BW)
      N=(BB/RESL)+.001

c    In this example, BB=3.95; thus N=158 and the dimensions of INDEX
c    and MATRIX are 3950 and 24964 respectively as calculated by the
c    formula in the users guide.
c    It should be noted that the actual number of distinct traces in this
c    example is 419.

 2201 FORMAT(I4,3F12.6)
      CALL ROUTER(DX,DY,XM,YM,TYFE,TYFB,WK,IWK1,IWK2,IWK3,MATRIX,
     1INDEX,N)
      STOP
      END
      SUBROUTINE ROUTER(DX,DY,XM,YM,TYFE,TYFB,DSS,NNO,NOS,ISCA,
     1MATRIX,INDEX,N7)

C THIS SUBROUTINE IS THE EXECUTIVE OF THE ROUTING ALGORITHM

      DIMENSION X3(150),Y3(150),JFDD(20)
      DIMENSION NNO(1),DSS(1),NSTAT(200),X2(150),Y2(150)
```

```
      DIMENSION NOS(1),JI(2,2),IOR(4),XS(150),YS(150)
      DIMENSION DX(1),DY(1),XM(1),YM(1),ISCA(1)
      DIMENSION ISIDE(100,2),TYFB(1),
     1TYFE(1),TYPO(2),AYER(2)
      INTEGER*2 MATRIX(N7,N7,2),INDEX(N7,50),ICC,IC
      COMMON/MAIN/JW,JWW,IDES(2),
     1ICC(50),IC(50),ITEM(2),RES,RESK,RESKK
      COMMON/DAVID/KT,PDW,N1,XLL(20),YLL(20),XUR(20),YUR(20),BW,
     1BH,NOO,NL1,NL2,RESL
      common/ICOT/ICOT
      COMMON/STTT/YFB,YFE,KMN
      COMMON/DAVE/LZ,KX
      COMMON/RICE/TYPE,FF,POW,kmm1,kmn1
      COMMON/JR/LM,JFD
      COMMON/COORD/XDES,YDES
      COMMON/EXTRA/XTRA,YTRA
      OPEN(UNIT=4,FILE='XY')
      TOLL=.00001
      RESL=RESL*.99999
      KD=KT
      TYPE=0
      NSURF=NOO
      AYER(1)=NL1*2
      AYER(2)=NL2*2
      PDW=PDW*1000
      RES=RESL
      AMAXX=-9999
      AMINX=99999
      AMAXY=-9999
      AMINY=9999
      DO 1544 J=1,KT
      AMINX=AMIN1(XM(J),DX(J),AMINX)
      AMAXX=AMAX1(XM(J),DX(J),AMAXX)
      AMAXY=AMAX1(YM(J),DY(J),AMAXY)
      AMINY=AMIN1(YM(J),DY(J),AMINY)
 1544 CONTINUE
      TX=AMAXX-AMINX
      TY=AMAXY-AMINY
      BTX=(BW-TX)*.5
      BTY=(BH-TY)*.5
      DX1=AMINX-BTX
      DY1=AMINY-BTY
      IDX=DX1/RES
      DX1=IDX*RES
      IDY=DY1/RES
      DY1=IDY*RES
      DO 1533 J=1,KT
      XM(J)=XM(J)-DX1
      YM(J)=YM(J)-DY1
      DX(J)=DX(J)-DX1
      DY(J)=DY(J)-DY1
 1533 CONTINUE
      I12=0
      TZRO=0
      ZRO=0
      TFED=0
      SFED=0
```

```
      KKK=1
      MMV=0
      DMMV=0
      XMAX=-9999
      XMIN=99999
      YMAX=-9999
      YMIN=99999
      DO 7001 KMN=1,NSURF
      kmn1=kmn
      PNB=AYER(KMN)/2
      NB=PNB
      NBB=NB
      IF(KMN.NE.1)GO TO 8009
      DO 1788 J=1,KT
 1788 NOS(J)=0
      LLP1=0
      KJ1=0
      KL1=0
      KD1=0
      LLJ=0
      TD=0
      TD1=0
      LKK=0
 8009 CONTINUE
      POW=1
      DO 1904 KMM=1,NB
      kmm1=kmm
      SC=RES
      JI(1,1)=1
      JI(2,1)=BW/RESL
      JI(1,2)=1
      JI(2,2)=BH/RESL

c  THIS CODE OPENS AND CLOSES PLOT DEVICE

c      IF((KMM.NE.1).OR.(KMN.NE.1))CALL CLOSE(IDEV)
      CALL SETUP(IDEV)
C
      KKK=IDEV
 1766 CONTINUE

C THIS CODE CONSTRUCTS INDEXES WHICH DETERMINE THE INITIAL AVAILABILITY OF
C ALL THE ROWS AND COLUMNS ON THE PCB.

      nn=N7
      NNC=NN
      DO 1600 J=1,2
      DO 1601 L=1,NN
      DO 1602 M=1,NN
      MATRIX(M,L,J)=NN-M
 1602 CONTINUE
 1601 CONTINUE
 1600 CONTINUE
 2200 FORMAT(20I3)
      KBB=0
      DO 1234 KB=1,50
 1234 NSTAT(KB)=0
```

```
      TK=KMN

C THIS CODE INITIALIZES THE DIRECTORIES ONLY AT THE ORIGINS AND CORRESPONDING
C TARGETS.

      DO 6677 J=1,KT
      MX=XM(J)/RES
      NX=YM(J)/RES
      MXX=DX(J)/RES
      NXX=DY(J)/RES
      NC=-2
      IF((TYFB(J).EQ.0.).OR.(TYFB(J).EQ.TK))GO TO 110
      IF((TYFE(J).EQ.0.).OR.(TYFE(J).EQ.TK))GO TO 112
      IF((TYFB(J).EQ.TYFE(J)).AND.(TYFE(J).NE.TK))GO TO 6677
  110 CONTINUE
      MATRIX(MX,NX,1)=NC
      MATRIX(NX,MX,2)=NC
  112 CONTINUE
      MATRIX(MXX,NXX,1)=NC
      MATRIX(NXX,MXX,2)=NC
 6677 CONTINUE

C INITIALIZATION DONE

 6600 CONTINUE
      IF((KMN.NE.1).OR.(KMM.NE.1))GO TO 3400
      JFF=0
      JFD=0
      AMAXA=0
      TJE=0
      TJEE=0
      IXC=0
      SUM=0
      DO 5599 LY=1,KD
      XZ=XM(LY)-DX(LY)
      YZ=YM(LY)-DY(LY)
      AREA=ABS(XZ*YZ)
      AMAXA=AMAX1(AMAXA,AREA)
      DDX=(XZ**2+YZ**2)
      IF(DDX.EQ.0.)GO TO 5599
      IXC=IXC+1
      SUM=SUM+DDX**.5
      TJEE=TJEE+1
      IF(AREA.EQ.0.)TJE=TJE+1
      XMAX=AMAX1(XMAX,XM(LY),DX(LY))
      XMIN=AMIN1(XMIN,XM(LY),DX(LY))
      YMAX=AMAX1(YMAX,YM(LY),DY(LY))
      YMIN=AMIN1(YMIN,YM(LY),DY(LY))
      JFD=JFD+1
      NNO(JFD)=LY
      IF(NSURF.GT.1)AREA=0
      DSS(JFD)=DDX+2.5*AREA
 5599 CONTINUE

c  HERE THE SCALING IS DONE FOR THE SCREEN

      cvv1=ymax-ymin
```

```
      CVV=xmax-xmin
      cvv=amax1(cvv,cvv1)
      cvv=cvv*1.3
      sd=cvv

C   ENDS SCALING

      SUM=SUM/IXC
      YMID=(YMAX+YMIN)/2
      XMID=(XMAX+XMIN)/2
      KS=JFD
      CALL VSRTR(DSS,KS,NNO)
      ITB=0
      IF(TJE/TJEE.GE..7)ITB=1
      PER=TJE/TJEE
C
C     N1 IS THE NO. OF AREAS THAT WILL BE UNROUTABLE!!!!!
C     THEREFORE, THESE AREAS ARE MADE TO BE UNROUTABLE BY INDICATING
C     THEIR OCCUPANCY IN THE DIRECTORY

      IF(N1.EQ.0)GO TO 3400
      YFE=0
      YFB=0
      JYY=0
      DO 4242 JJJ=1,N1
      K1=(YUR(JJJ)-YLL(JJJ))/RES
      K2=(XUR(JJJ)-XLL(JJJ))/RES
      NX=(YLL(JJJ)/RES)
      DO 170 JC=1,K1
      NX=NX+1
      MX=(XLL(JJJ)/RES)
      DO 160 JCC=1,K2
      MX=MX+1
      IZ=NX
      IU=MX
      MATRIX(MX,NX,1)=-2
      MATRIX(NX,MX,2)=-2
      CALL INIT(MX,NX,IU,IZ,KMM,MATRIX,INDEX,N7)
  160 CONTINUE
  170 CONTINUE
 4242 CONTINUE
 3400 CONTINUE

C  PROCESS IS COMPLETE

      JFDD(KMM)=JFD-LLP1
      U=2./(SD-SC)
      V=1.-(SD*U)
      AMAXX=-9999
      AMAXY=-9999
      AMINX=99999
      AMINY=99999

C  AT THIS POINT THE ENTIRE DIRECTORY IS NOW CORRECTED IN TERMS OF ITS
C  AVAILABILITY OF ROWS AND COLUMNS
C
      DO 180 NNX=1,KT
```

```
      YFB=TYFB(NNX)
      YFE=TYFE(NNX)
      AMAXX=AMAX1(XM(NNX),DX(NNX),AMAXX)
      AMAXY=AMAX1(YM(NNX),DY(NNX),AMAXY)
      AMINX=AMIN1(XM(NNX),DX(NNX),AMINX)
      AMINY=AMIN1(YM(NNX),DY(NNX),AMINY)
      INX=(XM(NNX)/RES)
      INY=(YM(NNX)/RES)
      IFX=(DX(NNX)/RES)
      IFY=(DY(NNX)/RES)
      CALL INIT(INX,INY,IFX,IFY,KMM,MATRIX,INDEX,N7)
  180 CONTINUE

C DIRECTORY IS NOW CURRENT AND READY FOR ACTUAL ROUTING

 2599 FORMAT(1X,15I5)
      PC=(AMAXX-AMINX)*(AMAXY-AMINY)/KT
      IF(IDEV.EQ.0)GO TO 2111
CXXXX
      IF((TYPE.EQ.-1).AND.(NSURF.NE.1))GO TO 3998
      IF((TYPE.EQ.-1.).AND.(KMM.NE.1))GO TO 2111
 3998 CONTINUE
      CALL GRID(SC,SD,KT,XM,YM,DX,DY,TYFB,TYFE,KMM)
 1257 CONTINUE
 2111 CONTINUE
      LLP=0
      IF(NSURF.NE.1)LQQ=6
      LQQ=8
      KZ=2
      I77=1
 1008 CONTINUE
      ICC(49)=LQQ
      DO 2501 KX=1,LQQ
      DO 2503 LZ=1,JFD
      INDEX(N7-1,50)=-1
      IK6=0
      XTRA=0
      YTRA=0
      LM=NNO(LZ)
      IF(NOS(LM).GE.1)GO TO 2503
      IF(KMM.GT.1)GO TO 1566
      IF((LZ.GE..99*JFD).AND.(KX.LE.6))GO TO 2501
 1566 CONTINUE
      XZ=XM(LM)-DX(LM)
      YZ=YM(LM)-DY(LM)
      IIX=ABS(XZ/RES)
      IIY=ABS(YZ/RES)
      IF(XZ.GT.YZ)JN7=1
      DIST=(XZ**2+YZ**2)**.5
      XOLD=XM(LM)
      YOLD=YM(LM)
      XOLD1=DX(LM)
      YOLD1=DY(LM)
      IF(KMN.EQ.1)GO TO 2700
      IF(TYFB(LM)+TYFE(LM).EQ.2)GO TO 2500
      GO TO 2800
 2700 CONTINUE
```

```
      IF(TYFB(LM)+TYFE(LM).EQ.4)GO TO 2500
 2800 CONTINUE
      RESKK=PDW/1000.
      ij=0
 3311 CONTINUE
      KHH=0
      DEX=XM(LM)-XMID
      DEX1=DX(LM)-XMID
      DEY=YM(LM)-YMID
      DEY1=DY(LM)-YMID
      DELX=XM(LM)-DX(LM)
      DELY=YM(LM)-DY(LM)
      ICC(50)=0
      IF((DELX*DELY.EQ.0.).AND.(LZ.LE..5*JFD))ICC(50)=1
      IF(ITB.EQ.0)GO TO 9866
      IF((DEX*DEX1.GE.0.).OR.(DEY*DEY1.GE.0.))GO TO 1166
      IF(KX.LT.5)GO TO 2500
 1166 CONTINUE
      IF((AMAXA.EQ.0.).OR.(LZ.LE..3*JFD))GO TO 6117
      IF((KX.eq.1).AND.(DELX*DELY.EQ.0.))GO TO 2500
 6117 CONTINUE
      IF((KX.LT.4).AND.(LZ.GE..9*JFD))GO TO 2500
      IF(DELX*DELY.EQ.0.)GO TO 9866
      IF((KX.eq.1).AND.(LZ.GE..5*JFD))GO TO 2500
 9866 CONTINUE
      AREA=DELY*DELX
      AREA=ABS(AREA)
      IF(AMAXA.EQ.0.)AMAXA=1
      AA=AREA/AMAXA
      BB=DSS(LZ)/DSS(JFD)
      IOR(1)=(XM(LM)/RES)
      IOR(2)=(YM(LM)/RES)
      IOR(3)=(DX(LM)/RES)
      IOR(4)=(DY(LM)/RES)
 5588 FORMAT(1X,10I6)
      XX=XM(LM)
      YY=YM(LM)
      VV=DX(LM)
      WW=DY(LM)
      TYPO(1)=TYFB(LM)
      TYPO(2)=TYFE(LM)
      IF((AA.LE..4).OR.(BB.LE..4))GO TO 4444
      IF(KX.NE.1)GO TO 1212
      IF((AA.GE..75).OR.(BB.GE..75))GO TO 9097
 1212 CONTINUE
      IF(KX.LE.3)GO TO 9097
 4444 CONTINUE

C THE MATRIX IS TEMPORARILY CORRECTED AT THE ORIGIN AND TARGET IN ORDER
C TO MAKE AVAILABLE THE ORIGIN AND TARGET OF THE CURRENT TRACE ATTEMPTED.
C OTHERWISE,THE TRACE WOULD NOT BE POSSIBLE AS THESE LOCATIONS ARE NOT
C AVAILABLE FOR OTHER TRACES

      DO 1700 L=1,2
      N=2*L-1
      MX=IOR(N)
      NX=IOR(N+1)
```

```
      IF(L.EQ.1)ITY=TYFB(LM)
      IF(L.EQ.2)ITY=TYFE(LM)
      NX1=NX-1
      NX2=NX+1
      MX1=MX-1
      MX2=MX+1
      J3=0
      DO 1702 J1=NX1,NX2
      J3=J3+1
      J4=0
      DO 1780 J2=MX1,MX2
      J4=J4+1
      MX=MX2-J4+1
      NX=NX2-J3+1
      IF((J4.EQ.2).AND.(J3.EQ.2))MATRIX(MX,NX,1)=0
      IF(ITY.NE.0)GO TO 1104
      IF((J4.NE.2).OR.(J3.NE.2))GO TO 1780
 1104 CONTINUE
      IF((MATRIX(MX,NX,1).EQ.-2).OR.(MATRIX(MX,NX,1).LE.-4))GO TO 1780
      EXP=0
      IF(MATRIX(MX,NX,1).EQ.-3)EXP=1
      IF(MATRIX(MX,NX,1).EQ.-3)MATRIX(MX,NX,1)=0
      IF(MATRIX(MX+1,NX,1).LT.0)GO TO 1301
      IF(MX.NE.JI(2,1))MATRIX(MX,NX,1)=MATRIX(MX+1,NX,1)+1
 1301 CONTINUE
      IF((J3.EQ.2).AND.(J4.EQ.3))MATRIX(MX,NX,1)=1
      M1=MX-1
      IF(M1.LE.0)GO TO 1401
      IF(MATRIX(M1,NX,1).LT.0)GO TO 1401
      LL=1
      IF(EXP.EQ.1.)LL=MATRIX(MX,NX,1)+1
      MATRIX(M1,NX,1)=LL
      LX= MATRIX(M1,NX,1)
      MM=M1-1
      IF((MM.LE.0).OR.(MATRIX(MM,NX,1).LT.0.))GO TO 1401
      CALL FREE(1,MM,1,NX,MM,III,III1,MATRIX,INDEX,N7)
      DO 9 J=III1,MM
    9 MATRIX(J,NX,1)=MATRIX(J,NX,1)+LX
 1401 CONTINUE
      IF((J4.EQ.2).AND.(J3.EQ.2))MATRIX(NX,MX,2)=0
      IF(ITY.NE.0)GO TO 1105
      IF((J4.NE.2).OR.(J3.NE.2))GO TO 1780
 1105 CONTINUE
      IF((MATRIX(NX,MX,2).EQ.-2).OR.(MATRIX(NX,MX,2).LE.-4))GO TO 1780
      EXP=0
      IF(MATRIX(NX,MX,2).EQ.-3)EXP=1.
      IF(MATRIX(NX,MX,2).EQ.-3)MATRIX(NX,MX,2)=0
      IF(MATRIX(NX+1,MX,2).LT.0)GO TO 1302
      IF(NX.NE.JI(2,2))MATRIX(NX,MX,2)=MATRIX(NX+1,MX,2)+1
 1302 CONTINUE
      IF((J3.EQ.3).AND.(J4.EQ.2))MATRIX(NX,MX,2)=1
      M2=NX-1
      IF(M2.LE.0)GO TO 1402
      IF(MATRIX(M2,MX,2).LT.0)GO TO 1402
      LL=1
      IF(EXP.EQ.1.)LL=MATRIX(NX,MX,2)+1
      MATRIX(M2,MX,2)=LL
```

```
      LX=MATRIX(M2,MX,2)
      MM=M2-1
      IF((MM.LE.0).OR.(MATRIX(MM,MX,2).LT.0.))GO TO 1402
      CALL FREE(2,MM,1,MX,MM,III,III1,MATRIX,INDEX,N7)
      DO 8 J=III1,MM
    8 MATRIX(J,MX,2)=MATRIX(J,MX,2)+LX
 1402 CONTINUE
 1780 CONTINUE
 1702 CONTINUE
 1700 CONTINUE
 1777 CONTINUE
C
C THE TEMPORARY CORRECTION OF DIRECTORY IS DONE
C
 5411 FORMAT(1X,5I10)
      L5=0
      ISW=0
 1235 CONTINUE
      XDES=DX(LM)
      YDES=DY(LM)
      ICOT=0
      IF(KX.Lt.6)GO TO 5551
      IF(KX.EQ.LQQ)GO TO 5551
      DXX=DX(LM)-XM(LM)
      DYY=DY(LM)-YM(LM)
      KZ=2
      IF((DXX.EQ.0.).OR.(DYY.EQ.0.))GO TO 5551
      GG=AMAX1(XM(LM),DX(LM))
      DX(LM)=(JI(2,1)-KZ)*RES
      IF(XM(LM).EQ.GG)DX(LM)=(JI(1,1)+KZ)*RES
      IF(DY(LM).GE.YM(LM))S=(((JI(2,2)-KZ)*RES)+DY(LM))/2.
      IF(DY(LM).LT.YM(LM))S=(((JI(1,2)+KZ)*RES)+DY(LM))/2.
      DY(LM)=S
      ICOT=1
 5551 CONTINUE
      IDES(1)=(DX(LM)/RES)
      IDES(2)=(DY(LM)/RES)
      IDY=IDES(2)
      IDX=IDES(1)
      XTRA=0
      YTRA=0
      IF(KX.NE.LQQ)GO TO 1919
      IF((IIX.LE.3).OR.(IIY.LE.3))GO TO 1919
      IF((MATRIX(IDX,IDY+1,1).GE.0).AND.(MATRIX(IDY+1,
     1IDX,2).GE.0))GO TO 1918
      IF((MATRIX(IDX+1,IDY,1).GE.0).AND.(MATRIX(IDY,
     1IDX+1,2).GE.0))GO TO 1917
      IF((MATRIX(IDX,IDY-1,1).GE.0).AND.(MATRIX(IDY-1,
     1IDX,2).GE.0))GO TO 1920
      IF((MATRIX(IDX-1,IDY,1).GE.0).AND.(MATRIX(IDY,IDX-1,2).
     1GE.0))GO TO 1921
      GO TO 1919
 1920 CONTINUE
      XTRA=0
      YTRA=RES
      IDES(2)=IDY-1
      GO TO 1919
```

```
 1921 CONTINUE
      XTRA=RES
      YTRA=0
      IDES(1)=IDX-1
      GO TO 1919
 1917 CONTINUE
      YTRA=0
      XTRA=-RES
      IDES(1)=IDX+1
      GO TO 1919
 1918 CONTINUE
      XTRA=0
      YTRA=-RES
      IDES(2)=IDY+1
 1919 CONTINUE
C
C THE ACTUAL ALGORITHM BEGINS(SEE THEORETICAL PAPER)
C
      ITEM(1)=(XM(LM)/RES)
      ITEM(2)=(YM(LM)/RES)
      IW=IABS(ITEM(1)-IDES(1))
      IWW=IABS(ITEM(2)-IDES(2))
      IF(L5.EQ.0)GO TO 104
      IF(IW.GE.IWW)GO TO 1055
      GO TO 100
  104 CONTINUE
      IF(IW.GE.IWW)GO TO 100
 1055 CONTINUE
      JW=2
      JWW=1
      GO TO 101
  100 CONTINUE
      JW=1
      JWW=2
  101 J=0
      K0=ITEM(JW)
      K01=ITEM(JWW)
      DO 102 JD=1,50
  102 IC(JD)=1
      JS=1
      IF(KX.GT.1)JS=2
      KM=50
      IF(KX.EQ.2)KM=30
      IF((KX.EQ.3).OR.(KX.EQ.4))KM=40
      IF(DELX*DELY.EQ.0.)KM=30
      IF(KX.EQ.1)KM=15
      IF((KX.GE.5).AND.(NSURF.GT.1))KM=100
      IF(KX.EQ.1)IU=2
      IF(KX.GE.2)IU=(KX-1)*2+3
      IF(KX.GE.4)JS=3
      IF(KX.LT.5)GO TO 1155
      JS=10
      IF((DELX*DELY.NE.0.).AND.(LZ.LE..7*JFD))JS=6
      IF((DELX*DELY.EQ.0.).AND.(LZ.LE..2*JFD))JS=8
 1155 CONTINUE
      IF(KX.LT.7)GO TO 1199
      IF(DELX*DELY.EQ.0.)GO TO 1199
```

```
      JS=10
      KM=100
      IU=40
 1199 CONTINUE
      IF((LLP.LE..7*JFDD(KMM)).OR.(KX.LE.5))GO TO 1002
      IF(DELX*DELY.EQ.0.)GO TO 1002
      IU=MAX0(30,IU)
      KM=MAX0(KM,100)
 1002 CONTINUE
 1005 CONTINUE
      IF(ISTAT.EQ.-99)JS=300
      II6=MAX0(IDES(JWW),K01)
      II7=MIN0(IDES(JWW),K01)
      II=MIN0(II6+JS,JI(2,JWW))
      II1=MAX0(II7-JS,JI(1,JWW))
 2005 CONTINUE
      JJ=JWW
      MAX=9999
      MIN=0
      NR=K0+IJ
      MR=K0-IJ

C AT A SPECFIC POINT ON THE MAJOR AXIS, POINTS(PIXELS) ARE CHOSEN IN A NESTED
C FASHION ON THE MINOR AXIS.

      DO 1760 N=MR,NR
      CALL FREE(JJ,II,II1,N,K01,III,III1,MATRIX,INDEX,N7)
      MAX=MIN0(III,MAX)
      MIN=MAX0(III1,MIN)
 1760 CONTINUE
      III=MAX
      III1=MIN
      IF((K01.EQ.III).AND.(K01.EQ.III1))GO TO 105
      J=J+1
      IF(J.GE.IU)GO TO 9097
      CALL FILL(III,III1,K0,K01,J,MATRIX,INDEX,N7)
 1010 CONTINUE
      LE=IC(J)
      LT=ICC(J)
      KK=INDEX(LT+1,J)

C KK REPRESENTS THE NEXT PIXEL ON THE MINOR AXES WITH RESPECT TO GIVEN
C "J(i)" ON THE MAJOR AXIS, AS IN TP3639.PDF.
C THE ACTUAL SEARCH FOR A SOLUTION STARTS HERE.

      DO 220 I=LE,LT
      JQ=IC(J)
      CALL ROUTE(KK,JQ,K0T,J,ISTAT,XM,YM,DX,DY,MATRIX,INDEX,N7)
      IF(ISTAT.LT.0)GO TO 1312
      IF((K0T.GE.JI(2,JW)).OR.(K0T.LE.JI(1,JW)))GO TO 321
      IF(ISTAT.EQ.0)GO TO 1312
      IF((ISTAT.GE.JI(2,JWW)).OR.(ISTAT.LE.JI(1,JWW)))GO TO 321
 1312 CONTINUE

C IF SUCCESSFUL, WE ARE DONE AND THE DIRECTORY IS CORRECTED TO REFLECT THE
C NEW TRACE. ISTAT > 0 INDICATES TOTAL SUCCESS.
```

```
      IF(ISTAT.GT.0)GO TO 1090

C IF NOT SUCCESSFUL, THE ALGORITHM PROCEEDS AS DESCRIBED IN THE PAPER
C THAT IS, ISTAT=0 INDICATES PARTIAL SUCCESS, WHILE ISTAT < 0 INDICATES
C THAT NO PROGRESS WAS MADE

      I99=1
      IF(PC.LT..06)I99=2
      IF((ISTAT.EQ.-2).AND.(KX.LT.LQQ))GO TO (1101,321),I99
      IF(RES.GT..049)GO TO 9911
      IF((ISTAT.LT.0).AND.(kx.le.1))GO TO 321
 9911 CONTINUE
      IF(PC.LT..02)GO TO 1214
      IF((KX.GE.2).OR.(DIST.GE..45))GO TO 1214
      IF((J.GE.2).OR.(IC(J).GE.2))GO TO 1101
 1214 CONTINUE
      IF((KX.LE.6).OR.(J.LE.6))GO TO 1113
      go to 321
 1113 CONTINUE
      IF(ISTAT.LT.0)GO TO 9922
      IF((KX.GE.LQQ).AND.(J.GT.1))GO TO 321
      IF((KX.GE.LQQ).AND.(J.EQ.1))ISTAT=-99
      IF((KX.GE.LQQ).AND.(J.EQ.1))GO TO 221
 9922 CONTINUE
      IUT=5
      IF(DIST.lt.1.5)IUT=3
      IF((ISTAT.EQ.0).AND.(KX.GE.iut))GO TO 221
      IK6=IK6+1
      IF((IK6.GT.20).AND.(KX.LT.3))GO TO 1101
      IF((IK6.GT.100).AND.(ISTAT.EQ.0))GO TO 221
      IF((IK6.GT.800.).AND.(ISTAT.LT.0))GO TO 1101
      IF((KX.GT.3).OR.(XZ*YZ.EQ.0.))GO TO 7088
      IF((ISTAT.LT.0).AND.(IC(J).GT.4))GO TO 1101
 7088 CONTINUE
      I88=1
      IF(KX.GE.7)I88=2
      IF((J.GE.9).AND.(LZ.GE..4*JFD))GO TO (1101,321),I88
      IF((KX.EQ.1).AND.(ITB.EQ.1))GO TO 1101
      IF(KX.EQ.1)GO TO 321
      IF((KHH.EQ.0).AND.(IC(J).GE.3))GO TO 9199
      IF(J.LE.3)GO TO 2133
      IF((XZ*YZ.EQ.0).AND.(ISTAT.EQ.0))GO TO 9198
 2133 CONTINUE
      IF((J.GT.1).AND.(KX.LE.2))GO TO 1101
      GO TO 9198
 9199 CONTINUE
      IF((J.GT.1).AND.(KX.eq.2))GO TO 321
 9198 CONTINUE
      CON1=.82
      IF(KX.GT.2)CON1=.7
      con2=.9
      IF((LZ.le..3*JFD).and.(j.gt.3))GO TO 1213
      IF((XZ*YZ.EQ.0).and.(J.LE.6))GO TO 321
      IF(DIST.LE.1.5)GO TO 3212
      IF(J.LT.5)CON2=.93
 1213 CONTINUE
      IF(IC(J).GE.7)GO TO 3212
```

```
      IF((LLP.GE..35*JFDD(KMM)).AND.(XZ*YZ.NE.0.))CON1=.84
 3212 CONTINUE
      IF((J.eq.1).or.(KHH.GE.3))GO TO 1112
      IF((KX.LE.3).AND.(LZ.GE.CON1*JFD))GO TO 1101
 1112 CONTINUE
      IF((KX.eq.4).AND.(LZ.GE.CON2*JFD))GO TO 1101
 3211 CONTINUE
      IF((IC(J).GE.6).AND.(KX.LE.3))GO TO 1101
      IF(KMM.GT.1)GO TO 8755
      IF(PC.GT..1)GO TO 7999
      iff=3
      IF(ic(j).ge.4)IFF=4
      IF((ISTAT.EQ.0).AND.(KX.GE.iff))GO TO 221
 8755 CONTINUE
      IF((XZ*YZ.EQ.0).or.(LLP.LE..10*JFDD(KMM)))GO TO 7899
      IF((J.GT.3).AND.(KX.LT.4))GO TO 1101
 7899 CONTINUE
      IF(KX.GE.7)GO TO 7999
 7999 CONTINUE
      IF(KMM.EQ.1)GO TO 321
      IF((J.GT.3).AND.(KX.EQ.4))GO TO 321
      IF((J.GT.6).AND.(KX.EQ.5))GO TO 1101
      IF((ISTAT.EQ.0).AND.(KX.EQ.2))GO TO 321
      IF(ISTAT.EQ.0)GO TO 221
  321 CONTINUE
      IC(J)=IC(J)+1
  220 CONTINUE

C AT THIS JUNCTURE,THE PRECEDING GENERATION IS ADDRESSED(ANCESTOR)
C AND THE ALGORITHM IS ADVANCED AS BEFORE.

  222 J=J-1
      KHH=KHH+1
      IF((J.LE.5).OR.(KMM.NE.1))GO TO 214
      KQ=5
      IF((KHH.GE.3).AND.(KX.LE.KQ))GO TO 9097
  214 CONTINUE
      IF(KHH.GE.KM)J=0
  219 CONTINUE
      IF(J.LE.0)GO TO 1090

C HERE THE NEXT PIXEL OF THE MINOR AXIS IS ADDRESSED AT THE PRESENT
C GENERATION

      IC(J+1)=1
      IF(IC(J)+1.GT.ICC(J))GO TO 222
      IC(J)=IC(J)+1
      GO TO 1010
  221 CONTINUE

c HERE THE NEXT GENERATION IS SOUGHT WHERE THE NEXT VALUE ON THE
C MAJOR AXIS IS K0T="J(i+1)"(this is the symbolic notation in the
c theoretical paper)

      K0=K0T
      K01=INDEX(IC(J),J)
      GO TO 1005
```

```
  105 CONTINUE
      MN=IDES(JW)-K0
      LL=999
      LL1=999
      IF(MN.GE.0)LL=K0+MN
      IF(MN.LT.0)LL1=K0+MN
      IF(LL.EQ.999)LL=K0
      IF(LL1.EQ.999)LL1=K0
      MAX=9999
      MIN=0
      NR=K01+IJ
      MR=K01-IJ
      DO 1860 N=MR,NR
      CALL FREE(JW,LL,LL1,N,K0,LLL,LLL1,MATRIX,INDEX,N7)
      MAX=MIN0(LLL,MAX)
      MIN=MAX0(LLL1,MIN)
 1860 CONTINUE
      LLL=MAX
      LLL1=MIN
      IF((K0.NE.LLL).OR.(K0.NE.LLL1))GO TO 2115
      IF(J.GE.1)GO TO 219
      GO TO 1090
 2115 CONTINUE
      K00=K0
      IF(IDES(JW).EQ.K0)GO TO 222
      DO 1960 N=MR,NR
      IF(IDES(JW).GT.K0)I=MATRIX(K0,N,JW)
      IF(IDES(JW).LT.K0)I=MATRIX(K0-1,N,JW)
      IF(I.LE.0)GO TO 222
 1960 CONTINUE
      K0=K0+ISIGN(1,IDES(JW)-K0)

C PROCEED TO THE NEXT GENERATION AND SUBSEQUENTLY DETERMINE NEW
C PIXELS FOR THE MINOR AXIS.

      J=J+1
      ICC(J)=1
      IC(J)=1
      INDEX(1,J)=K01
      INDEX(2,J)=K00
      GO TO 2005
 1090 CONTINUE
 2388 FORMAT(1X,3I10)
      IF(J.GE.IU)GO TO 9097
      IF(J.GT.0)GO TO 1100
 1101 CONTINUE
 1236 CONTINUE
      ISW=ISW+1
      IF(KX.LT.6)GO TO 1616
      IF(ISW.EQ.2)GO TO 1717
      DX(LM)=VV
      DY(LM)=WW
      GO TO 1616
 1717 CONTINUE
      DX(LM)=XX
      DY(LM)=YY
 1616 CONTINUE
```

```
      XMM=XM(LM)
      YMM=YM(LM)
      XM(LM)=DX(LM)
      YM(LM)=DY(LM)
      DX(LM)=XMM
      DY(LM)=YMM

C HERE DECISIONS ARE MADE TO EITHER SWITCH MINOR AND MAJOR AXES OR
C SOURCES AND TARGETS.

      IF(ISW.EQ.2)GO TO 9098
      GO TO 1235
 9098 CONTINUE
      IF((IWW.EQ.0).OR.(IW.EQ.0))GO TO 9097
 9099 CONTINUE
      L5=L5+1
      IF(L5.EQ.2)GO TO 9097
      ISW=0
      GO TO 1235
 9097 CONTINUE
      YFB=TYFB(LM)
      YFE=TYFE(LM)

C RESET TARGET AND SOURCE TO BE UNUASABLE FOR OTHER TRACES.

      CALL INIT(IOR(1),IOR(2),IOR(3),IOR(4),KMM,MATRIX,INDEX,N7)
      XM(LM)=XX
      YM(LM)=YY
      DX(LM)=VV
      DY(LM)=WW
 5050 CONTINUE
      TKMN=KMN
      IF(KMM.LT.NBB)GO TO 2505
      IF((TYFB(LM).EQ.0.).AND.(KMN.EQ.NSURF))GO TO 2277
      IF(TYFB(LM).EQ.TKMN)GO TO 2277
      GO TO 2505
 2277 CONTINUE
      IF(KX.EQ.LQQ)LKK=LKK+1
      IF(KX.EQ.LQQ)NOS(LM)=-KMN
      GO TO 2505
 1100 CONTINUE
 1077 CONTINUE
 1012 CONTINUE
  506 FORMAT(1X,7I9)

C SUCCESS .... PROCEED TO MAKE THE PROPER ENTRANCE IN THE DIRECTORY
C TO REFLECT THAT TRACE AND CORRECT THE ENTIRE DIRECTORY. MOREOVER,
C PLACE THE ACTUAL CONNECTIONS IN THE FILE "XY" WHICH TAKES PLACE
C IN THE SUBROUTINE, "CONNECT".

      IF(INDEX(N7,50).EQ.1)I12=I12+1
      CALL DIRCTY(J,ISTAT,K0T,MATRIX,INDEX,N7)
      YFB=TYFB(LM)
      YFE=TYFE(LM)
      CALL INIT(IOR(1),IOR(2),IOR(3),IOR(4),KMM,MATRIX,INDEX,N7)
      CALL CONNECT(J,SC,SD,K0T,ISTAT,MATRIX,INDEX,N7)
      XM(LM)=XX
```

```
      YM(LM)=YY
      DX(LM)=VV
      DY(LM)=WW
 2504 CONTINUE
      LKK=LKK+1
 4500 CONTINUE
      LLP=LLP+1
      ISCA(LLP)=LM
      IF(ISW.NE.0)ISCA(LLP)=-LM
      NOS(LM)=1
 2505 CONTINUE
 2500 CONTINUE
      DX(LM)=XOLD1
      DY(LM)=YOLD1
      XM(LM)=XOLD
      YM(LM)=YOLD
 2503 CONTINUE
 2501 CONTINUE
 2502 CONTINUE
      INN=0
      DO 1870 J=1,KT
      INX=XM(J)/RES
      INY=YM(J)/RES
      IFX=DX(J)/RES
      IFY=DY(J)/RES
      MATRIX(INX,INY,1)=-2
      MATRIX(INY,INX,2)=-2
      MATRIX(IFX,IFY,1)=-2
      MATRIX(IFY,IFX,2)=-2
 1870 CONTINUE
      DO 1288 J=1,2
      DO 1289 L=1,NNC
      DO 1290 M=1,NNC
      IF(MATRIX(M,L,J).LE.-4)MATRIX(M,L,J)=-1
 1290 CONTINUE
 1289 CONTINUE
 1288 CONTINUE
 1905 CONTINUE
      INN=INN+1
      REWIND 4
      KJ=0
      IF(LLP.EQ.0)GO TO 7001
      IO=1
      IF(TYPE.EQ.1.)IO=2
      DO 1611 JG=1,IO
      KL=0
      ISW=0
      TQ=0
 5555 DO 3131 KP=1,1000
 2323 FORMAT(2F9.3)

C READS THE LINE-SEGMENTS OF THE TRACES

      READ(4,2323)XS(KP),YS(KP)
      IF(XS(KP).LE.-999.)GO TO 120
 3131 CONTINUE
  120 CONTINUE
```

```
      KP=KP-1
      TX=ABS(XS(1)-XS(KP))
      TY=ABS(YS(1)-YS(KP))
      NG=KP
  500 CONTINUE
      DO 16 J=1,NG
      IF(J+2.GT.NG)GO TO 19
      T=XS(J+1)-XS(J)
      T1=XS(J+2)-XS(J+1)
      S=YS(J+1)-YS(J)
      S1=YS(J+2)-YS(J+1)
      IF((T.EQ.0.).AND.(T1.EQ.0.))GO TO 18
      IF((S.EQ.0.).AND.(S1.EQ.0.))GO TO 18
      GO TO 16
   18 CONTINUE
      NN=J+2
      DO 17 JT=NN,NG
      XS(JT-1)=XS(JT)
      YS(JT-1)=YS(JT)
   17 CONTINUE
      NG=NG-1
      GO TO 500
   16 CONTINUE
   19 CONTINUE
      KP=NG
      KL=KL+1
      KF=IABS(ISCA(KL))
      TX=ABS(DX(KF)-XM(KF))
      TY=ABS(DY(KF)-YM(KF))
      TD=TD+TX+TY
      KFF=ISCA(KL)
      IBC=TYFB(KF)
      IEC=TYFE(KF)
      IF(KFF.GE.0)GO TO 3838
      IBC=TYFE(KF)
      IEC=TYFB(KF)
 3838 CONTINUE
      IJJ=0
      JKL=MOD(KL,7)
      IF(JKL.EQ.3)JKL=0

c     Determines the color of the trace

c      CALL JCOLOR(JKL)
 2222 CONTINUE
CXXXX
      IF(KL.EQ.LLP)ISW=1
      IOR(1)=0
      IOR(2)=0

C THE FOLLOWING CODE IS BEING EMPLOYED TO DETERMINE IF A PARTICULAR
C LINE-SEGMENT OF THIS TRACE IS IN FACT NOT CROSSED BY ANY OTHER TRACE
C ON THE ALTERNATE LAYER. IF NOT, THEN NO VIA IS NECESSARY.

      DO 1305 JX=1,2
      LN=0
      IF((JX.EQ.2).AND.(IOR(2).GE.999))GO TO 1305
```

```
      DO 1300 JH=1,KP
      IF(JH.EQ.KP)GO TO 1300
      XD=XS(JH)-XS(JH+1)
      YD=YS(JH)-YS(JH+1)
      IF(JX.NE.1)GO TO 9090
      TD1=TD1+ABS(XD)+ABS(YD)
 9090 CONTINUE
      IF(XD.EQ.0.)GO TO 1901
      JW=1
      JWW=2
      NX=(YS(JH)+TQ)/RES
      JM=(XS(JH)+TQ)/RES
      JMM=(XS(JH+1)+TQ)/RES
      GO TO 1902
 1901 CONTINUE
      JW=2
      JWW=1
      JM=(YS(JH)+TQ)/RES
      JMM=(YS(JH+1)+TQ)/RES
      NX=(XS(JH+1)+TQ)/RES
 1902 CONTINUE
      J1=MIN0(JM,JMM)
      J2=MAX0(JM,JMM)
 3301 FORMAT(1X,8I8)
      IF(JW.EQ.2)GO TO 6888
      A1=XS(JH)
      B1=XS(JH+1)
      GO TO 5990
 6888 CONTINUE
      A1=YS(JH)
      B1=YS(JH+1)
 5990 CONTINUE
      IF(J2-J1.LE.0)GO TO 2007
      NR=NX+IJJ
      MR=NX-IJJ
      A=XS(JH)-XS(JH+1)
      B=YS(JH)-YS(JH+1)
      DO 2006 N=MR,NR
      DO 150 LH=J1,J2
      IF((LH.LE.J1).OR.(LH.GE.J2))GO TO 150
      IF(IJJ.EQ.0)GO TO 1866
      IF((A.LE.0.).AND.(B.LE.0.))GO TO 1888
      IF((JH.EQ.1).AND.(LH.LE.J1+IJJ))GO TO 150
      IF((JH.EQ.KP-1).AND.(LH.GE.J2-IJJ))GO TO 150
      GO TO 1999
 1888 CONTINUE
      IF((JH.EQ.1).AND.(LH.GE.J2-IJJ))GO TO 150
      IF((JH.EQ.KP-1).AND.(LH.LE.J1+IJJ))GO TO 150
 1999 CONTINUE
      IF((JH.EQ.KP-1).OR.(JH.EQ.1))GO TO 1866
      IF((LH.GE.J2-IJJ).OR.(LH.LE.J1+IJJ))GO TO 150
 1866 CONTINUE
      IF(N.LE.1)GO TO 150
      L1=MATRIX(N-1,LH,JWW)
      L2=MATRIX(N,LH,JWW)
      L3=MATRIX(N+1,LH,JWW)
      IF(L2.GE.0)GO TO 150
```

```
      IF((L1.GE.0).OR.(L3.GE.0))GO TO 150
      GO TO 2266
  150 CONTINUE
 2006 CONTINUE
 2007 CONTINUE
      IF(JH.NE.1)GO TO 1300
      IF(JX.EQ.1)JWO=JWW
      IF(JX.EQ.2)JWO=JW
      IOR(JX+2)=JWO
      GO TO 1300
 2266 CONTINUE
      IF(JH.NE.1)GO TO 2999
      IOR(2)=9999
      JWO=JW
      IOR(JX+2)=JWO
      GO TO 1300
 2999 CONTINUE
      IF(JWO.EQ.JW)GO TO 1300
      LN=LN+1
      ISIDE(LN,JX)=JH
      JWO=JW
      IOR(JX)=IOR(JX)+1
 1300 CONTINUE
 1305 CONTINUE
      IF(IOR(1).LE.IOR(2))KJJ=IOR(1)
      IF(IOR(2).LE.IOR(1))KJJ=IOR(2)
      IF(KJJ.EQ.IOR(1))JWOO=IOR(3)
      IF(KJJ.EQ.IOR(2))JWOO=IOR(4)
      IF(INN.EQ.2)NSTAT(KJJ+1)=NSTAT(KJJ+1)+1
      KBB=MAX0(KBB,KJJ+1)
      KJ=KJ+KJJ
      IF(KJJ.EQ.IOR(1))LB=1
      IF(KJJ.EQ.IOR(2))LB=2
      I=0
      KG=KP-1
      IDD=MAX0(KP,IDD)
      LMM=1

C THIS CONCLUDES THE ARGUMENT THAT DETERMINES VIA PLACEMENT

      DO 1908 JX=1,KG
      IF(JX.NE.1)GO TO 1907
      I=I+1
      X2(I)=XS(1)
      Y2(I)=YS(1)
      GO TO 1908
 1907 CONTINUE
      XD=XS(JX-1)-XS(JX)
      YD=YS(JX-1)-YS(JX)
      XD1=XS(JX+1)-XS(JX)
      YD1=YS(JX+1)-YS(JX)
      IF((XD.EQ.0.).AND.(XD1.EQ.0.))GO TO 1908
      IF((YD.EQ.0.).AND.(YD1.EQ.0.))GO TO 1908
      TOL=res
      t7=2
      IF((ABS(XD)+ABS(YD).LE.t7*res).OR.(ABS(XD1)+ABS(YD1).LE.t7*res))
     1TOL=res/2
```

```
      IF(XD.EQ.0.)GO TO 2001
      I=I+1
      IF(LMM.GT.KJJ)GO TO 1899
      IF(ISIDE(LMM,LB).NE.JX)GO TO 1899
      ISIDE(LMM,LB)=I
      LMM=LMM+1
      X2(I)=XS(JX)
      Y2(I)=YS(JX)
      GO TO 1908
 1899 CONTINUE
      X2(I)=XS(JX)+SIGN(TOL,XD)
      Y2(I)=YS(JX)
      I=I+1
      X2(I)=XS(JX)
      Y2(I)=YS(JX)+SIGN(TOL,YD1)
      GO TO 1908
 2001 CONTINUE
      I=I+1
      IF(LMM.GT.KJJ)GO TO 1807
      IF(ISIDE(LMM,LB).NE.JX)GO TO 1807
      ISIDE(LMM,LB)=I
      LMM=LMM+1
      X2(I)=XS(JX)
      Y2(I)=YS(JX)
      GO TO 1908
 1807 CONTINUE
      TOLI=.015
      X2(I)=XS(JX)
      Y2(I)=YS(JX)+SIGN(TOLI,YD)
      I=I+1
      Y2(I)=YS(JX)
      X2(I)=XS(JX)+SIGN(TOLI,XD1)
 1908 CONTINUE
      I=I+1
      X2(I)=XS(KP)
      Y2(I)=YS(KP)
 2077 CONTINUE
 3077 CONTINUE
      X2(1)=XS(1)
      Y2(1)=YS(1)
      DO 4422 JX=1,I
      Y3(JX)=Y2(JX)
      X3(JX)=X2(JX)
      MX=(X3(JX)/RES)+1
      NX=(Y3(JX)/RES)+1
 4422 continue
      DO 1122 JX=1,I
      X2(JX)=X2(JX)*U+V
      Y2(JX)=Y2(JX)*U+V
 1122 CONTINUE

C IF IDEV=0, DO NOT PLOT

      IF(IDEV.EQ.0)GO TO 1133
      KHG=KJJ+1
      ISIDE(KHG,LB)=I
      LMB=1
```

```
      DO 1690 MN=1,KHG
      II=X3(LMB)/RES
      JJ=Y3(LMB)/RES
      LT=ISIDE(MN,LB)-LMB+1
      JD=MOD(JWOO,2)
      JN=JD+4

c     Determines the type of line(dashed/solid)

c      IF(MN.EQ.1)CALL JLSTYL(JD)
c
c     Moves to point with pen up

c      IF(MN.EQ.1)CALL JMOVE(X2(LMB),Y2(LMB))
c
      IS=-8
      IS1=-8
      G=Y3(LMB)-Y3(LMB-1)
      GG=X3(LMB)-X3(LMB-1)
      HH=X3(LMB+1)-X3(LMB)
      H=Y3(LMB+1)-Y3(LMB)
      IX=0
      IF((MN.NE.1).OR.(KMM.EQ.1))GO TO 3377
      IF(IBC.EQ.0)GO TO 3377
      IF(INN.EQ.1)GO TO 9797
 5577 CONTINUE
      C=.0075
      IF((MATRIX(II-1,JJ,1).LE.-2).OR.(MATRIX(II+1,JJ,1).LE.-2))
     1C=.015
      IF((MATRIX(JJ-1,II,2).LE.-2).OR.(MATRIX(JJ+1,II,2).LE.-2))
     1C=.015
 2477 FORMAT(1X,2I10)
      KW=0
      M1=MATRIX(II-1,JJ,1)
      M2=MATRIX(II+1,JJ,1)
      M3=MATRIX(JJ-1,II,2)
      M4=MATRIX(JJ+1,II,2)
 1799 CONTINUE
      IF(C.EQ..0075)GO TO 3166
      KW=1
      TFED=TFED+1
      C=.0075
      JWX=2
      IF(G.EQ.0.)JWX=1
      M=-SIGN(1.,G+GG)
      IF(JWX.EQ.1)IS=MATRIX(JJ,II+M,2)
      IF(JWX.EQ.1)IS1=MATRIX(JJ-1,II+M,2)
      IF(JWX.EQ.2)IS=MATRIX(II,JJ+M,1)
      IF(JWX.EQ.2)IS1=MATRIX(II-1,JJ+M,1)
      IF((IS.GE.0).OR.(IS1.GE.0))C=.015
      IF(ABS(G+GG).LE.RES)C=.015
      IF(C.EQ..015)IX=1
      IF(C.EQ..015)GO TO 3266
      JWX=2
      IF(G.EQ.0.)JWX=1
      M=SIGN(1.,H+HH)
      IF(JWX.EQ.1)IS=MATRIX(II,JJ+M,1)
```

```
      IF(JWX.EQ.1)IS1=MATRIX(II-1,JJ+M,1)
      IF(JWX.EQ.2)IS=MATRIX(JJ,II+M,2)
      IF(JWX.EQ.2)IS1=MATRIX(JJ-1,II+M,2)
      IF((IS.GE.0).OR.(IS1.GE.0))C=.015
      IF(C.EQ..015)IX=2
 3266 CONTINUE
      IF(IX.EQ.0)GO TO 3166
      IF(G.NE.0.)GO TO 3366
      IT=X3(LMB)/RES
      JT=Y3(LMB)/RES
      JT=JT+SIGN(1.,H)
      IF(IX.NE.1)GO TO 8800
      IG=X3(LMB)/RES
      JO=Y3(LMB)/RES
      IG=IG-SIGN(1.,G+GG)
      N1=MATRIX(JO-1,IG,2)
      N2=MATRIX(JO+1,IG,2)
      GO TO 2424
 8800 CONTINUE
      IG=IT
      JO=JT
      N1=MATRIX(IG+1,JO,1)
      N2=MATRIX(IG-1,JO,1)
 2424 CONTINUE
 1222 FORMAT(1X,F8.3)
      IF((N1.LE.-2).OR.(N2.LE.-2))GO TO 9900
      X0=X3(LMB)*U+V
      Y0=Y3(LMB)+SIGN(RES,H)
      Y0=Y0*U+V
C      IF(IX.EQ.1)CALL JLSTYL(JD)

C     Moves to point with pen down

C      CALL JDRAW(X0,Y0)
C
C      IF(IX.EQ.2)CALL JLSTYL(JD)
      GO TO 3166
 3366 CONTINUE
      IT=X3(LMB)/RES
      IT=IT+SIGN(1.,HH)
      JT=Y3(LMB)/RES
      IF(IX.NE.1)GO TO 7700
      IG=X3(LMB)/RES
      JO=Y3(LMB)/RES
      JO=JO-SIGN(1.,G+GG)
      N1=MATRIX(IG-1,JO,1)
      N2=MATRIX(IG+1,JO,1)
      GO TO 1414
 7700 CONTINUE
      IG=IT
      JO=JT
      N1=MATRIX(JO+1,IG,2)
      N2=MATRIX(JO-1,IG,2)
 1414 CONTINUE
      IF((N1.LE.-2).OR.(N2.LE.-2))GO TO 9900
      Y0=Y3(LMB)*U+V
      X0=X3(LMB)+SIGN(RES,HH)
```

```
      X0=X0*U+V
C       IF(IX.EQ.1)CALL JLSTYL(JD)
C       CALL JDRAW(X0,Y0)
C       IF(IX.EQ.2)CALL JLSTYL(JD)
 3166 CONTINUE
      C=.0075
      IF(IX.EQ.0)GO TO 9900
      IF(IX.NE.1)GO TO 9800
      XT=X3(LMB)*U+V
      YT=Y3(LMB)*U+V
      IF(G.EQ.0.)XT=(X3(LMB)-SIGN(RES,G+GG))*U+V
      IF(G.NE.0.)YT=(Y3(LMB)-SIGN(RES,G+GG))*U+V
      GO TO 3434
 9800 CONTINUE
      XT=X0
      YT=Y0
 3434 CONTINUE
      SFED=SFED+1
      IF((M1.EQ.-6).OR.(M2.EQ.-6))SFED=SFED+1
      IF((M3.EQ.-6).OR.(M4.EQ.-6))SFED=SFED+1
 3535 CONTINUE

C     Draws approximate circle with 5 points

C       CALL JCIRCL(XT,YT,0.,C,5)
      X2(LMB)=X0
      Y2(LMB)=Y0
C       CALL JMOVE(X0,Y0)
      MATRIX(II,JJ,1)=-1
      MATRIX(JJ,II,2)=-1
      XE=(XT-V)/U
      YE=(YT-V)/U
      I1=XE/RES
      J1=YE/RES
      MATRIX(I1,J1,1)=-4
      MATRIX(J1,I1,2)=-4
      GO TO  4747
 9900 CONTINUE
      C=.0075
      XT=X3(LMB)*U+V
      YT=Y3(LMB)*U+V
C       CALL JDRAW(XT,YT)
C       CALL JCIRCL(XT,YT,0.,C,5)
C       CALL JLSTYL(JD)
      X2(LMB)=XT
      Y2(LMB)=YT
      IF(KW.EQ.0)GO TO 4747
      IF((M1.EQ.-4).OR.(M2.EQ.-4))GO TO 2020
      IF((M3.NE.-4).AND.(M4.NE.-4))GO TO 4747
 2020 CONTINUE
      MATRIX(II,JJ,1)=-6
      MATRIX(JJ,II,2)=-6
      GO TO 4747
 9797 CONTINUE
      MATRIX(II,JJ,1)=-4
      MATRIX(JJ,II,2)=-4
      GO TO 4747
```

```
 3377 CONTINUE
      IF(MN.EQ.1)GO TO 4747
      IF(INN.EQ.1)GO TO 7979
      GO TO 5577
 7979 CONTINUE
      MATRIX(II,JJ,1)=-4
      MATRIX(JJ,II,2)=-4
 4747 CONTINUE
      JA=JD+1
      IF(IO.EQ.1)GO TO 6665
      IF(JA.NE.JG)GO TO 6666
 6665 CONTINUE
      IF(INN.EQ.1)GO TO 6666
      IF(KHG.EQ.MN)GO TO 7777
      LC=LMB+LT-1
      G=Y3(LC)-Y3(LC-1)
      GG=X3(LC)-X3(LC-1)
      GX=GG+G
      GXX=ABS(GX)
      RESS=AMIN1(GXX,RES,.015)
      IF(G.EQ.0.)X2(LC)=(X3(LC)-SIGN(RESS,GX))*U+V
      IF(G.NE.0.)Y2(LC)=(Y3(LC)-SIGN(RESS,GX))*U+V
 7777 CONTINUE

c     Draws LT points in the arrays X2 AND Y2 begining at LMB

c      CALL JPOLY(X2(LMB),Y2(LMB),LT)

 6666 CONTINUE
CXXXXX
      LL=LMB+LT-1
      LMB=ISIDE(MN,LB)
      JWOO=JWOO+1
 1690 CONTINUE
      X2(LL)=X3(LL)*U+V
      Y2(LL)=Y3(LL)*U+V
      X2(1)=X3(1)*U+V
      Y2(1)=Y3(1)*U+V
      XXC=ABS(X3(LL)-XM(KF))
      XXD=ABS(X3(LL)-DX(KF))
      YYC=ABS(Y3(LL)-YM(KF))
      YYD=ABS(Y3(LL)-DY(KF))
c      IF((XXC.GT.toll).AND.(XXD.GT.toll))CALL JCIRCL(
c     1X2(LL),Y2(LL),0.,.0075,5)
c      IF((YYC.GT.toll).AND.(YYD.GT.toll))CALL JCIRCL(
c     1X2(LL),Y2(LL),0.,.0075,5)
      IF(KMM.NE.1)GO TO 1132
c      IF((IEC.NE.KMN).AND.(IEC.GT.0))CALL JCIRCL(X2(LL),Y2(LL),0.,
c     1.0075,5)
c      IF((IBC.NE.KMN).AND.(IBC.GT.0))CALL JCIRCL(X2(1),Y2(1),0.,.0075,5)
c      GO TO 1133
 1132 CONTINUE
c      IF(IEC.GT.0)CALL JCIRCL(X2(LL),Y2(LL),0.,.0075,5)
c      IF(IBC.GT.0)CALL JCIRCL(X2(1),Y2(1),0.,.0075,5)
 1133 CONTINUE
      IF(ISW.EQ.1)GO TO 1906
      GO TO 5555
```

```
 1906 CONTINUE
      IF(INN.NE.2)GO TO 1905
      IF(IO.EQ.1)GO TO 1611
      IF(JG.EQ.2)GO TO 1611
C      CALL CLOSE(IDEV)
      CALL SETUP(IDEV)
      CALL GRID(SC,SD,KT,XM,YM,DX,DY,TYFB,TYFE,KMM)
      REWIND 4
 1611 CONTINUE
      KJ1=KJ1+KJ
      KL1=KL1+KL
 1800 CONTINUE
      REWIND 4
      IF((TYPE.EQ.-1.).AND.(NSURF.GT.KMN))GO TO 1900
      IF((TYPE.EQ.-1.).AND.(KMM.NE.NB))GO TO 1900
      IF((KKK.EQ.0).AND.(KMM.NE.NB))GO TO 1900
      IF((KKK.EQ.0).AND.(KMN.NE.NSURF))GO TO 1900
C      IF((KMM.EQ.NB).AND.(KMN.EQ.NSURF))CALL CLOSE(IDEV)
      IF((KMM.EQ.NB).AND.(KMN.EQ.NSURF))GO TO 1900
 1900 CONTINUE
  766 format(1X,2I10)
      LLP1=LLP1+LLP
      IF(JFD.EQ.LLP1)go to 7001
 1904 CONTINUE
 7001 CONTINUE
C      IF((IDEV.EQ.1).AND.(KMM1.NE.NB))CALL CLOSE(IDEV)
      IF(IDEV.EQ.1)GO TO 2211
      IF(TD.EQ.0.)GO TO 2211
      TDF=TD/LLP1
      TDD=TD1/TD
      S=0
      LB=0
      DO 1228 KH=1,KD
      IF((NOS(KH).LE.0).OR.(NOS(KH).EQ.1000))NOS(KH)=0
      S=S+NOS(KH)
 1228 CONTINUE
 4553 FORMAT(1X,2I10,F8.3)
      S1=S
      GB=KD1
      GB1=GB
      IF(LLJ.EQ.0)LLJ=1
      GB=GB/LLJ
      GG=KJ1+I12
      GG=GG/KL1
  700 format(/2X,42HAVERAGE NUMBER OF FEEDTHROUGHS PER TRACE =,f8.4)
      WRITE(7,700)GG
      WRITE(7,711)TDD
 711  FORMAT(/2X,38H(ACTUAL DISTANCE/MANHATTAN DISTANCE) =,F8.4)
      LKK=MIN0(KD,LKK)
      S=S/LKK
      GBB=IDD
      SLK=LKK
 7222 FORMAT(/2X,26HPERCENTAGE OF COMPLETION =,F7.4)
      GMMV=MMV
      WRITE(7,7222)S
      NTRR=S1
      WRITE(7,7223)NTRR
```

```
 7223 FORMAT(/2X,35HACTUAL NUMBER OF COMPLETED TRACES =,I5)
 7224 FORMAT(/2X,34HTOTAL NUMBER OF ATTEMPTED TRACES =,I5/)
      WRITE(7,7224)LKK
 2211 CONTINUE
      RETURN
      END
      SUBROUTINE FREE(JJ,II,II1,K0,K01,III,III1,MATRIX,INDEX,N7)

C THIS SUBROUTINE DETERMINES THE AVAILABILITY OF THE ROW OR COLUMN
C IN QUESTION WITHIN THE LIMITS REQUESTED.

      DIMENSION ITWO(16)
      INTEGER*2 MATRIX(N7,N7,2),INDEX(N7,50),ICC,IC
      DATA (ITWO(I),I=1,16)/1,2,4,8,16,32,64,128,256,512,1024,2048,
     14096,8192,16384,32768/
      COMMON/MAIN/JW,JWW,IDES(2),
     1ICC(50),IC(50),ITEM(2),RES,RESK,RESKK
      COMMON/DAVE/LZ,KD
 2100 FORMAT(1X,7I7)
      IHH=0
      III1=K01
      II3=MATRIX(K01,K0,JJ)
      IF(II3.LT.0)II3=0
      III=MIN0(II3+K01,II)
      EN=K01-II1
      IF(EN.LE.0.)GO TO 1007
      I=(ALOG10(EN)/ALOG10(2.))+2
      L=II1
 1001 CONTINUE
      M=MATRIX(L,K0,JJ)
      IF(M.LT.0)M=0
      M=M-(K01-L)
      IF((M.GE.0).AND.(L.EQ.II1))GO TO 1003
 1002 I=I-1
      IF(I.EQ.0)GO TO 1003
      L=L-ISIGN(ITWO(I),M)
      GO TO 1001
 1003 CONTINUE
      III1=L+1
      IF(MATRIX(L,K0,JJ).GE.K01-L)III1=L
 1007 RETURN
      END
      SUBROUTINE FILL(III,III1,K0,K01,J,MATRIX,INDEX,N7)

C THIS SUBROUTINE CONSTRUCTS THE PIXELS ON THE MINOR AXIS GIVEN A
C FIXED POINT ON THE MAJOR AXIS.

      INTEGER*2 MATRIX(N7,N7,2),INDEX(N7,50),ICC,IC
      COMMON/MAIN/JW,JWW,IDES(2),
     1ICC(50),IC(50),ITEM(2),RES,RESK,RESKK
      IG=1
      DO 1000 JX=1,1000
      K=1
      IG=IG+1
      M=0
      M1=0
      IF(III.LT.K01+JX)GO TO 110
```

```
      M=1
      INDEX(IG,J)=K01+JX
  110 IF(III1.GT.K01-JX)GO TO 1001
      M1=1
      IF(M.EQ.0)K=0
      INDEX(IG+K,J)=K01-JX
      IG=IG+K
 1001 CONTINUE
      IF(M+M1.EQ.0)GO TO 1002
 1000 CONTINUE
 1002 CONTINUE
      IG=IG-1
      ICC(J)=IG
      INDEX(1,J)=K01
      INDEX(IG+1,J)=K0
      RETURN
      END
      SUBROUTINE ROUTE(KK,JQ,K0T,J,ISTAT,XM,YM,DX,DY,MATRIX,INDEX,N7)

C THIS SUBROUTINE ACTUALLY TRIES TO COMPLETE A TRACE INDEPENDENTLY
C OF WHERE IT IS EN ROUTE AND EMPLOYS THE SEARCH BY THE METHOD INDICATED
C IN THE SYMBOLIC STATEMENTS IN THE THEORETICAL PAPER.

      DIMENSION XM(1),YM(1),DX(1),DY(1)
      INTEGER*2 MATRIX(N7,N7,2),INDEX(N7,50),ICC,IC
      COMMON/MAIN/JW,JWW,IDES(2),
     1ICC(50),IC(50),ITEM(2),RES,RESK,RESKK
      COMMON/ICOT/ICOT
      COMMON/DAVE/ LZ,KD
      COMMON/JR/LM,JFD
      COMMON/COORD/XDES,YDES
      INDEX(N7,50)=0
      DXX=XM(LM)-DX(LM)
      DYY=YM(LM)-DY(LM)
      JB2=ICC(49)
      JMM=ICC(50)
      JJ1=INDEX(JQ,J)
      LB=0
      TX=RES-(.5*RESKK)+.001
      IJ=RESK/TX
C      IJ=RESK*1.5/RES
      IJ=1
      IF(RES-RESKK.GE.0)IJ=0
      IJ=RESKK/RES
      ij=0
      K0TT=K0T
      JJ=JW
      IF(IDES(JJ).GE.KK)GO TO 100
      II=KK
      II1=IDES(JJ)
      GO TO 101
  100 CONTINUE
      II=IDES(JJ)
      II1=KK
  101 CONTINUE
      K1=INDEX(JQ,J)
      KK1=K1
```

```
      NR=K1+IJ
      MR=K1-IJ
      MAX=9999
      MIN=0
      DO 2106 N=MR,NR
      CALL FREE(JJ,II,II1,N,KK,III,III1,MATRIX,INDEX,N7)
      MIN=MAX0(MIN,III1)
      MAX=MIN0(MAX,III)
 2106 CONTINUE
      III=MAX
      III1=MIN
      IF((III.EQ.IDES(JJ)).OR.(III1.EQ.IDES(JJ)))GO TO 109
      IF((III.GT.KK).OR.(III1.LT.KK))GO TO 110
  113 CONTINUE
      ISTAT=-1
      GO TO 111
  110 CONTINUE
      ISTAT=0
      IF(III.GT.KK)K0T=III
      IF(III1.LT.KK)K0T=III1
      IF((KD.NE.JB2).OR.(J.NE.1))GO TO 9922
      KMB=IABS(KK-K0T)
      IF(KMB.LE.1)GO TO 9922
      KSAVE=K0T
      KQ=MIN0(KK,K0T)
      IMAX=-99999
      DO 600 IJH=1,KMB
      KQ=KQ+1
      MATR=MATRIX(JJ1,KQ,JWW)
      IF((MATR.LE.IMAX).OR.(MATR.LT.0))GO TO 600
      KSAVE=KQ
      IMAX=MATR
  600 CONTINUE
      K0T=KSAVE
 9922 CONTINUE
      K0YY=K0T
      GO TO 111
  109 CONTINUE
      LB=1
      K0YY=K0T
      IF(III.EQ.IDES(JJ))K0T=III
      IF(III1.EQ.IDES(JJ))K0T=III1
      JJ=JWW
      J1=INDEX(JQ,J)
      IF(IDES(JJ).GE.J1)GO TO 200
      II=J1
      II1=IDES(JJ)
      GO TO 201
  200 CONTINUE
      II=IDES(JJ)
      II1=J1
  201 CONTINUE
      K1=IDES(JW)
      NR=K1+IJ
      MR=K1-IJ
      MIN=0
      MAX=9999
```

```
      DO 2206 N=MR,NR
      CALL FREE(JJ,II,II1,N,J1,III,III1,MATRIX,INDEX,N7)
      MIN=MAX0(MIN,III1)
      MAX=MIN0(III,MAX)
 2206 CONTINUE
      III=MAX
      III1=MIN
      IA=IABS(III-IDES(JJ))
      IB=IABS(III1-IDES(JJ))
      K0YY=K0T
      IF((IB.LE.1).OR.(IA.LE.1))GO TO 112
      IF(KD.EQ.JB2)GO TO 115
      IA=IDES(JW)-KK
      IF(IABS(IA).LE.1)GO TO 115
      K0T=IDES(JW)-ISIGN(1,IA)
      ISTAT=0
      GO TO 111
  115 CONTINUE
      ISTAT=-1
      GO TO 111
  112 CONTINUE
      ISTAT=IDES(JJ)
  111 CONTINUE

C IF IT DETERMINED THAT A TRACE CANNOT BE COMPLETED, THEN IT TRIES TO
C COMPLETE IT PIGGY-BACK. (THAT IS,IT WILL TERMINATE ON AN EXISTING
C TRACE WHOSE DESTINATION IS THE SAME.)

      IF(ISTAT.GT.0)GO TO 122
      IF(KD.LT.3)GO TO 121
      IF((KD.EQ.JB2).AND.(ISTAT.GT.0))GO TO 122
      IF((JMM.NE.0).AND.(KD.GE.6))GO TO 121
      IF((ISTAT.GT.0).AND.(KD.GE.6))ISTAT=0
      IF(ISTAT.NE.0)GO TO 121
      I1=MAX0(K0YY,KK)
      J1=MIN0(K0YY,KK)
      DO 151 JN=J1,I1
      I=0
      DO 991 KC=1,3
      KF=KK1-2+KC
      LX=MATRIX(KF,JN,JWW)
      IF(LX.GE.-3)GO TO 151
      LXD=IABS(LX)-3
      IF((XDES.EQ.XM(LXD)).AND.(YDES.EQ.YM(LXD)))I=I+1
      IF((XDES.EQ.DX(LXD)).AND.(YDES.EQ.DY(LXD)))I=I+1
      IF(I.NE.KC)GO TO 151
  991 CONTINUE
      ISTAT=KK1
      K0T=JN
      INDEX(N7,50)=1
      GO TO 121
  151 CONTINUE
      IF(LB.EQ.0)GO TO 121
      JT=IABS(KK-IDES(JW))-1
      IF(JT.LE.0)GO TO 121
      IF(KK.LT.K1)MJ=-1
      IF(KK.GT.K1)MJ=1
```

```
      JT=JT+1
      IL=0
      DO 999 JK=1,JT
      IL=IL+1
      LQ=K1+(JK-1)*MJ
      MR=LQ-IJ
      NR=LQ+IJ
      MIN=0
      MAX=9999
      DO 888 N=MR,NR
      CALL FREE(JJ,II,II1,N,JJ1,III,III1,MATRIX,INDEX,N7)
      MIN=MAX0(MIN,III1)
      MAX=MIN0(III,MAX)
  888 CONTINUE
      III=MAX
      III1=MIN
      IF((III.EQ.KK1).AND.(III1.EQ.KK1))GO TO 121
      IF(III.EQ.KK1)KKM=III1
      IF(III1.EQ.KK1)KKM=III
      I1=MAX0(KKM,KK1)
      J1=MIN0(KKM,KK1)
      DO 152 JN=J1,I1
      I=0
      DO 998 KC=1,3
      KF=LQ-2+KC
      LX=MATRIX(KF,JN,JW)
      IF(LX.GE.-3)GO TO 152
      LDX=IABS(LX)-3
      IF((XDES.EQ.XM(LDX)).AND.(YDES.EQ.YM(LDX)))I=I+1
      IF((XDES.EQ.DX(LDX)).AND.(YDES.EQ.DY(LDX)))I=I+1
      IF(I.NE.KC)GO TO 152
  998 CONTINUE
      ISTAT=JN
      K0T=LQ
      INDEX(N7,50)=1
      GO TO 121
  152 CONTINUE
  999 CONTINUE
  121 CONTINUE

C JUDGMENTS ARE USED TO ACCEPT OR REJECT A COMPLETED TRACE.
C IF THE SOLUTION IS DEEMED TO BE TOO CONVOLUTED AT TOO EARLY
C AN EFFORT, IT WILL REJECT THE SOLUTION UNTIL THE LAST POSSIBLE
C MOMENT OR UNTIL A SIMPLER SUBSEQUENT SOLUTION IS FOUND.

      IF((ISTAT.GT.0).AND.(DXX*DYY.EQ.0))GO TO 122
      IF((DXX*DYY.NE.0).AND.(KD.GE.6))GO TO 122
      IF((KD.GE.6).AND.(ISTAT.GT.0))GO TO 122
      IF(KD.GE.5)GO TO 129
      IF((ISTAT.GT.0).AND.(J.GT.4))ISTAT=-1
  129 CONTINUE
      IF(JMM.EQ.1)GO TO 122
      IJN=IABS(KK-K0T)
      IF(J.GE.3)GO TO 125
      KJJ=3
      IF(DXX*DYY.NE.0)KJJ=2
      IF((IJN.GT.3).OR.(KD.GT.KJJ))GO TO 125
```

```
      IF(ISTAT.GT.0)ISTAT=-1
  125 CONTINUE
      IF(J.LE.2)GO TO 122
      IF((KD.EQ.JB2).AND.(ISTAT.GT.0))GO TO 122
      IF((ISTAT.EQ.0).AND.(IJN.LE.3))ISTAT=-1
  122 CONTINUE
      IF(ISTAT.LT.0)GO TO 177
      JY=RESKK/RES
      IF(JY.EQ.0)GO TO 177
      JYY=2*JY+1
      IF(INDEX(N7-1,50).GE.0)GO TO 1260
      IF((KK1.EQ.ITEM(2)).or.(KK1.EQ.ITEM(1)))GO TO 1260
      IF(J.eq.1)GO TO 1260
      KK2=INDEX(1,J)
      DO 661 I6=1,JYY
      NJ0=(I6-1)-JY
      IF(NJ0.EQ.0)GO TO 661
      N1=MATRIX(KK+NJ0,KK2-1,JW)
      N2=MATRIX(KK+NJ0,KK2,JW)
      N3=MATRIX(KK+NJ0,KK2+1,JW)
      IF((N1.LT.0).OR.(N2.LT.0))ISTAT=-1
      IF(N3.LT.0)ISTAT=-1
  661 CONTINUE
 1260 CONTINUE
      IF(ISTAT.LT.0)GO TO 178
      IHQ=0
      IF((KK1.NE.ITEM(1)).AND.(KK1.NE.ITEM(2)))GO TO 223
      IF((KK.NE.ITEM(1)).AND.(KK.NE.ITEM(2)))GO TO 223
      IHQ=1
  223 CONTINUE
      IST=ISTAT
      JY=RESKK/RES
      IF(JY.EQ.0)GO TO 177
      JYY=2*JY+1
      DO 666 I6=1,JYY
      NJ0=(I6-1)-JY
      N1=MATRIX(K0T+NJ0,KK1,JW)
      N2=MATRIX(KK1+1,K0T+NJ0,JWW)
      N3=MATRIX(KK1-1,K0T+NJ0,JWW)
      N4=MATRIX(KK+NJ0,KK1,JW)
      N5=MATRIX(KK1+1,KK+NJ0,JWW)
      N6=MATRIX(KK1-1,KK+NJ0,JWW)
C   CHANGES HERE
      IJT=0
      IF((IST.EQ.KK1).AND.(INDEX(N7,50).EQ.1))IJT=1
      IF((IJT.EQ.1).AND.(NJ0.EQ.0))GO TO 222
  788 CONTINUE
      IF((N1.Lt.0).OR.(N2.LT.0))ISTAT=-1
      IF(N3.LT.0)ISTAT=-1
  222 CONTINUE
      IF(IHQ.EQ.1)GO TO 666
      IF((N4.LT.0).OR.(N5.LT.0))ISTAT=-1
      IF(N6.LT.0)ISTAT=-1
  666 CONTINUE
      IF(ISTAT.EQ.-1)GO TO 178
      IF(INDEX(N7,50).EQ.0)GO TO 177
      IF(ISTAT.EQ.KK1)GO TO 177
```

```
      DO 667 I6=1,JYY
      NJ0=(I6-1)-JY
      IF(NJ0.EQ.0)GO TO 667
      N1=MATRIX(ISTAT+NJ0,K0T,JWW)
      N2=MATRIX(K0T+1,ISTAT+NJ0,JW)
      N3=MATRIX(K0T-1,ISTAT+NJ0,JW)
      IF((N1.LT.0).OR.(N2.LT.0))ISTAT=-1
      IF(N3.LT.0)ISTAT=-1
      IF(ISTAT.EQ.-1)GO TO 178
  667 CONTINUE
  178 CONTINUE
  177 CONTINUE
      if(istat.le.0)go to 127
      IF((ICOT.EQ.1).AND.(INDEX(N7,50).NE.1))ISTAT=-2
  127 continue
      INDEX(N7-1,50)=ISTAT
      RETURN
      END
      SUBROUTINE DIRCTY(J,ISTAT,K0T,MATRIX,INDEX,N7)
C
C THIS PROGRAM UPDATES THE DIRECTORY(MATRIX) AFTER A SOLUTION.
C
      INTEGER*2 MATRIX(N7,N7,2),INDEX(N7,50),ICC,IC
      COMMON/MAIN/JW,JWW,IDES(2),
     1ICC(50),IC(50),ITEM(2),RES,RESK,RESKK
      COMMON/DAVE/LZ,KD
      COMMON/RICE/TYPE,FF,POW,kmm,kmn
      COMMON/JR/LM,JFD
  888 FORMAT(1X,6I8)
      IQ=-(LM+3)
      TX=RES-(.5*RESKK)+.001
      IJ=RESK/TX
      IJ=1
      IF(RES-RESKK.GE.0)IJ=0
      IJ=RESKK/RES
      IJ=0
      IF(POW.EQ.2.)IJ=0
      N=J
      DO 1 I=1,N
      MT=INDEX(IC(I),I)
      MAX=MAX0(MT,INDEX(1,I))
      MIN=MIN0(MT,INDEX(1,I))
      KG=INDEX(ICC(I)+1,I)
      MIN=MAX0(MIN-1,1)
      LK=IC(I)
      IF(IC(I).EQ.1)GO TO 15
      NR=KG+IJ
      MR=KG-IJ
      DO 2406 NW=MR,NR
      DO 2 M=MIN,MAX
    2 MATRIX(M,NW,JWW)=IQ
      MINN=MIN
      MINN=MAX0(MIN-1,1)
      JJ=JWW
      CALL FREE(JJ,MINN,1,NW,MINN,III,III1,MATRIX,INDEX,N7)
      DO 3 M=III1,MINN
      IX=MIN-M
```

```
    3 IF(IX.LT.MATRIX(M,NW,JWW))MATRIX(M,NW,JWW)=IX
 2406 CONTINUE
   15 CONTINUE
      IF(I.EQ.N)GO TO 200
      KG1=INDEX(ICC(I+1)+1,I+1)
      MAX=MAX0(KG,KG1)
      MIN=MIN0(KG,KG1)
      MIN=MAX0(MIN-1,1)
      NR=MT+IJ
      MR=MT-IJ
      DO 2606 NW=MR,NR
      DO 20 M=MIN,MAX
   20 MATRIX(M,NW,JW)=IQ
      MINN=MAX0(MIN-1,1)
      JJ=JW
      CALL FREE(JJ,MINN,1,NW,MINN,III,III1,MATRIX,INDEX,N7)
      DO 30 M=III1,MINN
      IX=MIN-M
      IF(IX.LT.MATRIX(M,NW,JW))MATRIX(M,NW,JW)=IX
   30 CONTINUE
 2606 CONTINUE
      GO TO 1
  200 CONTINUE
      KG=INDEX(ICC(N)+1,N)
      MAX=MAX0(KG,K0T)
      MIN=MIN0(KG,K0T)
      IF(MIN.EQ.MAX)GO TO 3001
      MIN=MAX0(MIN-1,1)
      NR=MT+IJ
      MR=MT-IJ
      DO 2706 NW=MR,NR
      DO 2000 M=MIN,MAX
 2000 MATRIX(M,NW,JW)=IQ
 2300 FORMAT(1X,8I8)
      MINN=MIN-1
      MINN=MAX0(MINN,1)
      JJ=JW
      CALL FREE(JJ,MINN,1,NW,MINN,III,III1,MATRIX,INDEX,N7)
      DO 3000 M=III1,MINN
      IX=MIN-M
      IF(IX.LT.MATRIX(M,NW,JW))MATRIX(M,NW,JW)=IX
 3000 CONTINUE
 2706 CONTINUE
 3001 CONTINUE
      MAX=MAX0(ISTAT,MT)
      MIN=MIN0(ISTAT,MT)
      IF(MIN.EQ.MAX)GO TO 1
      MIN=MAX0(MIN-1,1)
      MINN=MAX0(MIN-1,1)
      NR=K0T+IJ
      MR=K0T-IJ
      DO 2806 NW=MR,NR
      DO 4000 M=MIN,MAX
 4000 MATRIX(M,NW,JWW)=IQ
      JJ=JWW
      CALL FREE(JJ,MINN,1,NW,MINN,III,III1,MATRIX,INDEX,N7)
      DO 5000 M=III1,MINN
```

```
      IX=MIN-M
      IF(IX.LT.MATRIX(M,NW,JWW))MATRIX(M,NW,JWW)=IX
 5000 CONTINUE
 2806 CONTINUE
    1 CONTINUE
      INX=ITEM(1)
      INY=ITEM(2)
      IFX=IDES(1)
      IFY=IDES(2)
      RETURN
      END
      SUBROUTINE CONNECT(J,SC,SD,K0T,ISTAT,MATRIX,INDEX,N7)

C THIS SUBROUTINE ACTUALLY CONSTRUCTS THE X,Y COORDINATES OF
C THE COMPLETED TRACE KNOWING THE INDEXES AND RESOLUTION OF THE GRID.

      INTEGER*2 MATRIX(N7,N7,2),INDEX(N7,50),ICC,IC
      COMMON/MAIN/JW,JWW,IDES(2),
     1ICC(50),IC(50),ITEM(2),RES,RESK,RESKK
      COMMON/DAVE/LZ,KD
      COMMON/EXTRA/XTRA,YTRA
      DIMENSION X(500),Y(500)
      BNINE=-9999
      BNINEE=99999
      U=2./(SD-SC)
      V=1.-(SD*U)
      I=0
      DO 1700 LX=1,J
      I=I+1
      IF(JW.EQ.2)GO TO 1800
      X(I)=INDEX(ICC(LX)+1,LX)*RES
      Y(I)=INDEX(1,LX)*RES
      IF(IC(LX).EQ.1)GO TO 1900
      I=I+1
      X(I)=X(I-1)
      Y(I)=INDEX(IC(LX),LX)*RES
 1900 CONTINUE
      IF(LX.EQ.J)GO TO 2000
      I=I+1
      X(I)=INDEX(ICC(LX+1)+1,LX+1)*RES
      Y(I)=Y(I-1)
      GO TO 1701
 1800 CONTINUE
      X(I)=INDEX(1,LX)*RES
      Y(I)=INDEX(ICC(LX)+1,LX)*RES
      IF(IC(LX).EQ.1)GO TO 1901
      I=I+1
      Y(I)=Y(I-1)
      X(I)=INDEX(IC(LX),LX)*RES
 1901 CONTINUE
      IF(LX.EQ.J)GO TO 2000
      I=I+1
      Y(I)=INDEX(ICC(LX+1)+1,LX+1)*RES
      X(I)=X(I-1)
      GO TO 1701
 2000 CONTINUE
      IF(JW.EQ.2)GO TO 4000
```

```
      I=I+1
      X(I)=K0T*RES
      Y(I)=Y(I-1)
      I=I+1
      X(I)=X(I-1)
      Y(I)=ISTAT*RES
      GO TO 1701
 4000 CONTINUE
      I=I+1
      X(I)=X(I-1)
      Y(I)=K0T*RES
      I=I+1
      X(I)=ISTAT*RES
      Y(I)=Y(I-1)
 1701 CONTINUE
 1700 CONTINUE
      DO 121 JH=1,I
      IF(JH.EQ.I)GO TO 10
      IF((X(JH).EQ.X(JH+1)).AND.(Y(JH).EQ.Y(JH+1)))GO TO 121
   10 CONTINUE
 3455 FORMAT(1X,2F8.3,I7)
      WRITE(4,2323)X(JH),Y(JH)
  121 CONTINUE
      IF((XTRA.EQ.0.).AND.(YTRA.EQ.0.))GO TO 33
      XQ=X(I)+XTRA
      YQ=Y(I)+YTRA
      WRITE(4,2323)XQ,YQ
   33 CONTINUE
      WRITE(4,2323)BNINE,BNINE
   22 CONTINUE
  900 CONTINUE
 2323 FORMAT(2F9.3)
      RETURN
      END
      SUBROUTINE GRID(SC,SD,KT,XM,YM,DX,DY,TYFB,TYFE,KMM)
C
C THIS SUBROUTINE DRAWS THE ENTIRE PRINTED CIRCUIT BOARD COMPLETE
C WITH ALL OF THE SOURCES AND TARGET PIXELS. SPECIAL SYMBOLS ARE
C USED IF THE SOURCE OR TARGET IS A SURFACE MOUNT.
C
      DOUBLE PRECISION U,V
      DIMENSION XM(1),YM(1),DX(1),DY(1),TYFB(1),TYFE(1)
      COMMON/STTT/YFB,YFE,KMN
      TKMN=KMN
      U=2./(SD-SC)
      V=1.-(SD*U)
      DO 177 JJ=1,KT
      IF((XM(JJ).NE.DX(JJ)).OR.(YM(JJ).NE.DY(JJ)))GO TO 712
      GO TO 177
  712 CONTINUE
      XX=XM(JJ)
      YY=YM(JJ)
      XX=XX*U+V
      YY=YY*U+V
      IF(KMM.EQ.1)GO TO 211
      IF(TYFB(JJ).NE.0.)GO TO 212
  211 CONTINUE
```

```
      IF((TYFB(JJ).NE.0.).AND.(TYFB(JJ).NE.TKMN))GO TO 212
      TTT=.01
      IF(TYFB(JJ).NE.0.)TTT=.005
      IF((TYFB(JJ)*TYFE(JJ).NE.0).AND.(TYFB(JJ).NE.TYFE(JJ)))TTT=.02
c      CALL JCIRCL(XX,YY,0.,TTT,5)
  212 CONTINUE
      XX=DX(JJ)
      YY=DY(JJ)
      IF(KMM.EQ.1)GO TO 2110
      IF(TYFE(JJ).NE.0)GO TO 177
  232 FORMAT(1X,4F8.3)
 2110 CONTINUE
      IF((TYFE(JJ).NE.0.).AND.(TYFE(JJ).NE.TKMN))GO TO 177
      XX=XX*U+V
      YY=YY*U+V
      TTT=.01
      IF(TYFE(JJ).NE.0.)TTT=.005
      IF((TYFB(JJ)*TYFE(JJ).NE.0.).AND.(TYFB(JJ).NE.TYFE(JJ)))TTT=.02
c      CALL JCIRCL(XX,YY,0.,TTT,5)
  177 CONTINUE
      RETURN
      END
      SUBROUTINE SETUP(IDEV)
C
      COMMON/RICE/TYPE,FF,POW,kmm,kmn

C     THIS ROUTINE INITIALIZE DI-3000 AND SETS THE HARDWARE OUTPUT
C     DEVICE FOR PLOTTING

      SCFE=1
      RATIO=SCFE
C
      SC=-1.
      SD=1.
      IDEV = 1
      WRITE(*,22)KMN,KMM
      WRITE (*,*) 'SELECT STATISTICS (0) OR PCB PLOT (1)'
   22 FORMAT(30HTHIS SOLUTION ADDRESSES SURFACE,1X,I2,1X,5HLAYER,1X,I2)
      READ *, IDEV
      IF(IDEV.NE.1)GO TO 9990

c  These next three statements are peculiar to the Tetronix 4107 terminal
c  Its function is to switch to plot mode from ascii mode.

c      PRINT *, '2J'
c      PRINT *, '
c      PRINT *, 'A0'
 9990 CONTINUE
c      CALL JBEGIN
c      IF (IDEV .EQ. 0) THEN
c         CALL JFSOPN (3,0,99,'METAFILE.DAT')
c      END IF
c      CALL JDINIT (IDEV)
c      CALL JDEVON (IDEV)
c      CALL JASPEK (IDEV,RATIO)
c      IF (RATIO .LT. 1.0) THEN
c         CALL JVSPAC (SC,SD,-RATIO,RATIO)
```

```
c          CALL JVPORT (SC,SD,-RATIO,RATIO)
c       ELSE
c          RATIO = 1./RATIO
c          CALL JVSPAC (-RATIO,RATIO,SC,SD)
c          CALL JVPORT (-RATIO,RATIO,SC,SD)
c       END IF
c       IF(IDEV.NE.1)GO TO 5656
C       WRITE(*,*) 'ENTER WINDOW BOUNDARIES (UMIN,UMAX,VMIN,VMAX)'
      UMIN=-1.
      UMAX=1.
      VMIN=-1.
      VMAX=1.
c       CALL JWINDO (UMIN,UMAX,VMIN,VMAX)
 5656 CONTINUE
c       CALL JFRAME
c       CALL JOPEN
      RETURN
      END
      SUBROUTINE CLOSE(IDEV)
C
C     THIS ROUTINE TERMINATES DI-3000 AND THE HARDWARE OUTPUT DEVICE
C
c       CALL JCLOSE
c       CALL JPAUSE (IDEV)
c       CALL JFRAME
c       CALL JDEVOF (IDEV)
c       CALL JDEND (IDEV)
c       CALL JEND
      IF(IDEV.NE.1)GO TO 9991
C       PRINT *, '
 9991 CONTINUE
      RETURN
      END
      SUBROUTINE INIT(INX,INY,IFX,IFY,KMM,MATRIX,INDEX,N7)
C
C THIS SUBROUTINE INITIALIZES THE DIRECTORY AT THE TARGET AND SOURCE.
C
      DIMENSION IOR(4)
      INTEGER*2 MATRIX(N7,N7,2),INDEX(N7,50),ICC,IC
      COMMON/MAIN/JW,JWW,IDES(2),
     1ICC(50),IC(50),ITEM(2),RES,RESK,RESKK
      COMMON/STTT/YFB,YFE,KMN
      IB=YFB
      IE=YFE
      IF(IE+IB.NE.3)GO TO 1010
      X=2
      GO TO 1800
 1010 CONTINUE
      IF(KMN.EQ.1)GO TO 1000
      IF((IB.EQ.1).AND.(IE.EQ.1))GO TO 1900
      GO TO 1001
 1000 CONTINUE
      IF((IB.EQ.2).AND.(IE.EQ.2))GO TO 1900
 1001 CONTINUE
      X=2
      IF((IB.EQ.0).OR.(IB.EQ.KMN))GO TO 1800
 1800 CONTINUE
```

```
      IOR(1)=INX
      IOR(2)=INY
      IOR(3)=IFX
      IOR(4)=IFY
      DO 1700 L=1,2
      IF((X.EQ.1.).AND.(L.EQ.1))GO TO 1700
      N=2*L-1
      MX=IOR(N)
      NX=IOR(N+1)
      IF(L.EQ.1)ITY=IB
      IF(L.EQ.2)ITY=IE
      MX1=MX-1
      MX2=MX+1
      NX1=NX-1
      NX2=NX+1
      DO 1602 J1=NX1,NX2
      DO 1780 J2=MX1,MX2
      MX=J2
      NX=J1
      IF((J2-MX1.EQ.1).AND.(J1-NX1.EQ.1))MATRIX(MX,NX,1)=-2
      IF(ITY.NE.0)GO TO 1104
      IF((J2-MX1.NE.1).OR.(J1-NX1.NE.1))GO TO 1780
 1104 CONTINUE
      IF((MATRIX(MX,NX,1).NE.-2).AND.(MATRIX(MX,NX,1).GT.-4))
     1MATRIX(MX,NX,1)=-3
      M1=MAX0(1,MX-1)
      IF(MATRIX(M1,NX,1).LT.0)GO TO 2000
      M10=MAX0(1,M1-1)
      MATRIX(M1,NX,1)=0
      IF(MATRIX(M10,NX,1).LE.0)GO TO 2000
      CALL FREE(1,M10,1,NX,M10,III,III1,MATRIX,INDEX,N7)
      DO 19 J=III1,M10
      IX=M1-J
   19 IF(IX.LT.MATRIX(J,NX,1))MATRIX(J,NX,1)=IX
 2000 CONTINUE
      M2=MAX0(1,NX-1)
      IF((J1-NX1.EQ.1).AND.(J2-MX1.EQ.1))MATRIX(NX,MX,2)=-2
      IF(ITY.NE.0)GO TO 1105
      IF((J1-NX1.NE.1).OR.(J2-MX1.NE.1))GO TO 1780
 1105 CONTINUE
      IF((MATRIX(NX,MX,2).NE.-2).AND.(MATRIX(NX,MX,2).GT.-4))
     1MATRIX(NX,MX,2)=-3
      IF(MATRIX(M2,MX,2).LT.0)GO TO 1780
      MATRIX(M2,MX,2)=0
      M11=MAX0(1,M2-1)
      IF(MATRIX(M11,MX,2).LE.0)GO TO 1780
      CALL FREE(2,M11,1,MX,M11,III,III1,MATRIX,INDEX,N7)
      DO 18 J=III1,M11
      IX=M2-J
   18 IF(IX.LT.MATRIX(J,MX,2))MATRIX(J,MX,2)=IX
 1780 CONTINUE
 1602 CONTINUE
 1700 CONTINUE
 1900 CONTINUE
      RETURN
      END
      SUBROUTINE VSRTR(A,LA,IR)
```

```
C   IMSL ROUTINE NAME   - VSRTR
C
C-----------------------------------------------------------------------
C
C   COMPUTER            - CDC/SINGLE
C
C   LATEST REVISION     - JANUARY 1, 1978
C
C   PURPOSE             - SORTING OF ARRAYS BY ALGEBRAIC VALUE -
C                           PERMUTATIONS RETURNED
C
C   USAGE               - CALL VSRTR (A,LA,IR)
C
C   ARGUMENTS    A      - ON INPUT, A CONTAINS THE ARRAY TO BE SORTED.
C                         ON OUTPUT, A CONTAINS THE SORTED ARRAY.
C                LA     - INPUT VARIABLE CONTAINING THE NUMBER OF
C                           ELEMENTS IN THE ARRAY TO BE SORTED.
C                IR     - VECTOR OF LENGTH LA.
C                         ON INPUT, IR CONTAINS THE INTEGER VALUES
C                           1,2,...,LA. SEE REMARKS.
C                         ON OUTPUT, IR CONTAINS A RECORD OF THE
C                           PERMUTATIONS MADE ON THE VECTOR A.
C
C   PRECISION/HARDWARE  - SINGLE/ALL
C
C   REQD. IMSL ROUTINES - NONE REQUIRED
C
C                           CONVENTIONS IS AVAILABLE IN THE MANUAL
C                           INTRODUCTION OR THROUGH IMSL ROUTINE UHELP
C
C   REMARKS      THE VECTOR IR MUST BE INITIALIZED BEFORE ENTERING
C                VSRTR.  ORDINARILY, IR(1)=1, IR(2)=2, ...,
C                IR(LA)=LA.  FOR WIDER APPLICABILITY, ANY INTEGER
C                THAT IS TO BE ASSOCIATED WITH A(I) FOR I=1,2,...,LA
C                MAY BE ENTERED INTO IR(I).
C
C   COPYRIGHT           - 1978 BY IMSL, INC. ALL RIGHTS RESERVED.
C
C   WARRANTY            - IMSL WARRANTS ONLY THAT IMSL TESTING HAS BEEN
C                           APPLIED TO THIS CODE.  NO OTHER WARRANTY,
C                           EXPRESSED OR IMPLIED, IS APPLICABLE.
C
C-----------------------------------------------------------------------
C
      DIMENSION A(1),IR(1)
C     EXTENDED DUMMY A,IR
C                                  SPECIFICATIONS FOR ARGUMENTS
C                                  SPECIFICATIONS FOR LOCAL VARIABLES
      INTEGER            IU(21),IL(21),I,M,J,K,IJ,IT,L,ITT
      REAL               T,TT,R
C                                  FIRST EXECUTABLE STATEMENT
      IF (LA.LE.0) RETURN
      M = 1
      I = 1
      J = LA
      R = .375
```

```
    5 IF (I.EQ.J) GO TO 45
      IF (R.GT..5898437) GO TO 10
      R = R+3.90625E-2
      GO TO 15
   10 R = R-.21875
   15 K = I
C                                   SELECT A CENTRAL ELEMENT OF THE
C                                   ARRAY AND SAVE IT IN LOCATION T
      IJ = I+(J-I)*R
      T = A(IJ)
      IT = IR(IJ)
C                                   IF FIRST ELEMENT OF ARRAY IS GREATER
C                                   THAN T, INTERCHANGE WITH T
      IF (A(I).LE.T) GO TO 20
      A(IJ) = A(I)
      A(I) = T
      T = A(IJ)
      IR(IJ) = IR(I)
      IR(I) = IT
      IT = IR(IJ)
   20 L = J
C                                   IF LAST ELEMENT OF ARRAY IS LESS THAN
C                                   T, INTERCHANGE WITH T
      IF (A(J).GE.T) GO TO 30
      A(IJ) = A(J)
      A(J) = T
      T = A(IJ)
      IR(IJ) = IR(J)
      IR(J) = IT
      IT = IR(IJ)
C                                   IF FIRST ELEMENT OF ARRAY IS GREATER
C                                   THAN T, INTERCHANGE WITH T
      IF (A(I).LE.T) GO TO 30
      A(IJ) = A(I)
      A(I) = T
      T = A(IJ)
      IR(IJ) = IR(I)
      IR(I) = IT
      IT = IR(IJ)
      GO TO 30
   25 IF (A(L).EQ.A(K)) GO TO 30
      TT = A(L)
      A(L) = A(K)
      A(K) = TT
      ITT = IR(L)
      IR(L) = IR(K)
      IR(K) = ITT
C                                   FIND AN ELEMENT IN THE SECOND HALF OF
C                                   THE ARRAY WHICH IS SMALLER THAN T
   30 L = L-1
      IF (A(L).GT.T) GO TO 30
C                                   FIND AN ELEMENT IN THE FIRST HALF OF
C                                   THE ARRAY WHICH IS GREATER THAN T
   35 K = K+1
      IF (A(K).LT.T) GO TO 35
C                                   INTERCHANGE THESE ELEMENTS
      IF (K.LE.L) GO TO 25
```

```
C                                     SAVE UPPER AND LOWER SUBSCRIPTS OF
C                                     THE ARRAY YET TO BE SORTED
      IF (L-I.LE.J-K) GO TO 40
      IL(M) = I
      IU(M) = L
      I = K
      M = M+1
      GO TO 50
   40 IL(M) = K
      IU(M) = J
      J = L
      M = M+1
      GO TO 50
C                                     BEGIN AGAIN ON ANOTHER PORTION OF
C                                     THE UNSORTED ARRAY
   45 M = M-1
      IF (M.EQ.0) RETURN
      I = IL(M)
      J = IU(M)
   50 IF (J-I.GE.11) GO TO 15
      IF (I.EQ.1) GO TO 5
      I = I-1
   55 I = I+1
      IF (I.EQ.J) GO TO 45
      T = A(I+1)
      IT = IR(I+1)
      IF (A(I).LE.T) GO TO 55
      K = I
   60 A(K+1) = A(K)
      IR(K+1) = IR(K)
      K = K-1
      IF (T.LT.A(K)) GO TO 60
      A(K+1) = T
      IR(K+1) = IT
      GO TO 55
      END
```

USER'S GUIDE

"Router" is the name of the software package that automatically routes signal traces given only the coordinates of the source and target.

The testpgm file contains the actual code. The router software initially expects the input to be consistent with the statistical summary rather than the plot. This expectation allows a match with the sample file to verify correct installation. In other words, the router software will ask the user to enter "0" for a statistical summary or "1" for a plot of actual signal traces. For purposes of installation, enter "0". Once the user has installed the plotting software for his particular installation, actual plotting can begin.

Definitions of the input variables in the calling sequence and the name-common variables follows, with dimensional size where appropriate.

First, the name-common and its corresponding variables:

```
common/david/nco,pdw,n1,xll(20),yll(20),xur(20),yur(20),bw,bh,noo,
nl1,nl2,resl
```

Each of the variables in name-common are now specifically defined (all dimensions are presumed to be in inches).

nco is the number of connections(If there are pins with no connections, then the source and target coordinates are the same and they will not contribute to the percentage of completions)

pdw is the pad diameter width

n1 is the number of unroutables rectangular areas.(Applicable only to boards which do not have surface mounts.)

xll(i) and yll(i) are the coordinates of the lower left corner of the ith unroutable area.

xur(i) and yur(i) are the coordinates of the upper right corner of the ith unroutable area.

bw and bh are the board width and height. These dimensions should include a border of at least 0.3 of an inch around the pins.

noo is the number of surfaces. A value of 1(one) means no surface mounts. a value of 2(two) means surface mounts are used.

nl1= layers associated with surface 1
nl2= layers associated with surface 2 (if noo is one, then nl2 is of no value. Also, each layer is two-sided.)

resl is the resolution of the grid and all values are assumed to be in inches.

The calling sequence is as follows:

```
call router(dx,dy,xm,ym,tyfe,tyfb,wk,iwk1,iwk2,iwk3,matrix,index,n)
```

where all the variables from dx through iwk3 are dimensioned the size equal to the number of connections. The meanings of this subset only, follows.

dx is the x coordinate of the target.

dy is the y coordinate of the target.

xm is the x coordinate of the source.

ym is the y coordinate of the source.

tyfe is one type of coordinates dx and dy.

tyfb is one type of coordinates xm and ym.

The type of each router is determined by the following codes where tyfe or tyfb may assume any of three values:

code=0 means the coordinate is a through-pin
code=1 means the coordinate is a surface mount on surface 1
code=2 means the coordinate is a surface mount on surface 2

For example, tyfb(1)=0 means the coordinate xm(1),ym(1) is a through-pin. If however tyfb(1)=2,then xm(1),ym(1) would be a surface mount on surface 2.....etc, etc.

Real work arrays and integer arrays:

wk is an undefined real work array.
iwk1, iwk2, and iwk3 are all integer work arrays.

The integer, n, is defined as:

n=[max(bh,bw)/resl]+.001

Thus, the dimension of the undefined integer-array matrix is n*n and the dimension of the undefined integer-array index is 25*n.

Note that (1) there is no input for the signal trace width, (2) it is assumed that the user will choose the resolution of the grid(resl) in such a way as to accommodate the signal traces and feed-throughs on two adjacent parallel tracks(rows/columns), and (3) it is assumed that all pin locations are multiples of the grid resolution.

The J-routines found in the router software can be replaced to correspond to the equivalent functions.

NASA Technical Paper 3639

September 1996

A Formal Algorithm for Routing Traces on a Printed Circuit Board

David R. Hedgley, Jr.

NASA Technical Paper 3639

1996

A Formal Algorithm for Routing Traces on a Printed Circuit Board

David R. Hedgley, Jr.
Dryden Flight Research Center
Edwards, California

National Aeronautics and Space Administration

Office of Management

Scientific and Technical Information Program

ABSTRACT

This paper addresses the classical problem of printed circuit board routing: that is, the problem of automatic routing by a computer other than by brute force that causes the execution time to grow exponentially as a function of the complexity. Most of the present solutions are either inexpensive but not efficient and fast, or efficient and fast but very costly. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based. This paper presents a formal algorithm for routing traces on a printed circuit board. The solution presented is very fast and efficient and for the first time speaks to the question eloquently by way of symbolic statements.

NOMENCLATURE

The following nomenclature do not appear in alphabetical order as each definition is predicated upon previous definitions.

μ the major axis (horizontal or vertical direction) where the sequence $\{J_i\}$ is associated with μ.

β the minor axis (horizontal or vertical direction) where the sequence $\{M_{k_i}\}$ is associated with β.

RM a rectangular matrix of evenly spaced grid points where a node is defined to be an arbitrary grid point belonging to RM.

RM_h RM represented as a sequence of horizontal lines composed of grid points.

RM_v RM represented as a sequence of vertical lines composed of grid points (fig. 1).

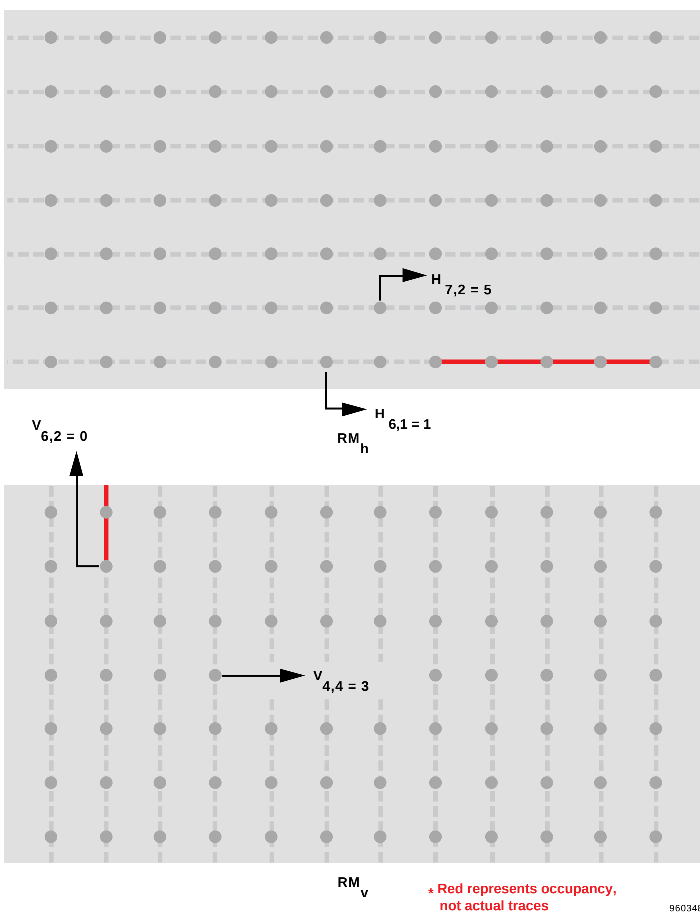

Figure 1. A representation of an internal matrix.

$H_{p,\,l}$ — the number of available nodes from left to right at the node, (p, l), that belongs to RM_h (fig. 1).

$V_{p,\,l}$ — the number of available nodes from bottom to top at the node, (p, l), that belongs to RM_v.

J_i — the i^{th} designated node on the major axis where sequence $\{J_i\}$ is monotonic and bounded. That is, if (IB, MB) and (IE, ME) are the initial and terminal nodes respectively, then $IB = J_1 > J_2 > J_3 > J_4 \ldots\ldots J_n = IE$, or $IB = J_1 < J_2 < J_3 < J_4 \ldots\ldots J_n = IE$. (These inequalities are consistent with the horizontal axis as the major axis. Otherwise, the sequence $\{J_i\}$ is bounded by MB and ME, and the inequalities will be consistent with the vertical axis as the major axis. Moreover, $|J_i - J_{i-1}|$ represents the number of available nodes between J_i and J_{i-1}. That is, the sequence $\{J_i\}$ is determined by availability (fig. 2). If IB = IE or MB = ME, then only one major axis (direction) exists.)

M_{k_i} — the k^{th} designated node on the minor axis that intersects the major axis at J_i, where M_{k_i} may not be monotonic or bounded for any i. The sequence $\{M_{k_i}\}$ is designated by the assignment of nodes in a cluster, if possible (fig. 2).

$N_i \geq 1$ — the number of nodes $(1 \leq k_i \leq N_i)$ on the minor axis associated with J_i. The M_{1_i} denotes the intersection node of the minor axis with the major axis at J_i.

$S(J_i, M_{k_i})$ — the discrete function that first interrogates the availability of a path belonging to either RM_h or RM_v, which is initiated at the node (J_i, M_{k_i}), in the major direction. If applicable, the function then subsequently interrogates the availability of a path in the minor direction if, and only if, $J_{i+1} = IE$ or ME. The values of $S(J_i, M_{k_i})$ are defined as follows:

- NA = No availability (not any space on the major axis is free).
- PA = Partial availability on the major axis.
- TA = Total availability (both axes are free to complete the trace).

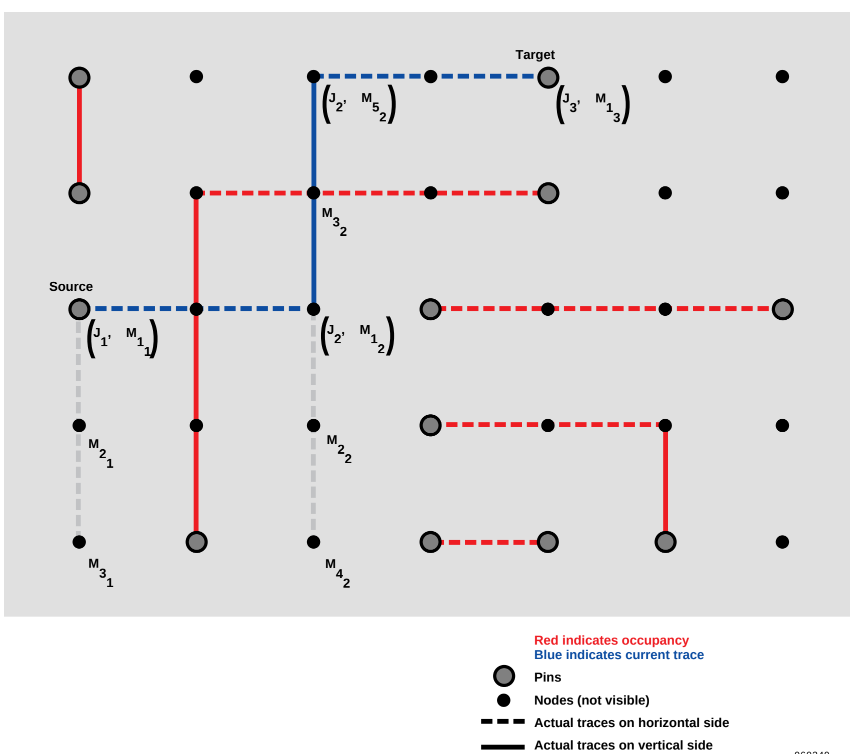

Figure 2. A representation of a typical trace.

NI no further interrogation, implying total success or total failure.

Definitions of Operations:

$C \leftrightarrows D$	the exchange of definitions of C and D.
$C \not\leftrightarrows D$	the exchange of definitions of C and D cannot take place.
$E[S(b, c)]$	evaluate $S(b, c)$.

INTRODUCTION

An automated solution to the printed circuit board problem that is both efficient and cost effective continues to be in great demand. Most of the present solutions[1–4] are either inexpensive but not efficient and fast, or efficient and fast but very costly. In either case, no solution exists that describes its algorithm symbolically with any degree of eloquence, nor are any algorithms sufficiently fast enough to facilitate parts placement for optimization. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based.

The purpose of this paper is to accomplish three objectives:

- To facilitate optimization of arrangement of parts (for example, resistors and capacitors) on printed circuit boards for more sophisticated and expensive systems.
- To make a very fast and efficient algorithm accessible at no cost for individual users or small companies.
- To serve as a pedagogical tool that has heuristic value by presenting a formal symbolic structure that describes the entire algorithm.

FORMAL SYMBOLIC STATEMENTS

The purpose of the following algorithm, presented as a list of symbolic statements, is to construct a path from the source node, (IB, MB), to the target node, (IE, ME). A typical path (fig. 2) is described by the sequence of nodes (J_1, M_{1_1}), (J_2, M_{1_2}), (J_2, M_{5_2}), and (J_3, M_{1_3}). In this case, $M_{1_1} = M_{1_2}$ and $M_{5_2} = M_{1_3}$. In general, if (J_i, M_{k_i}) is an arbitrary element in a sequence leading to a solution, then the next element in the sequence will be either $(J_{i+1}, M_{1_{(i+1)}})$ or $(J_i, M_{(k+u)_i})$, where $u \geq 0$. Thus, the algorithm is as follows:

(a) $N_i > 1, k_i < N_i,$ and $S(J_i, M_{k_i}) = NA \Rightarrow E[S(J_i, M_{(k+1)_i})]$

(b) $i > 1, N_i \geq 1, k_i = N_i,$ and $S(J_i, M_{k_i}) = NA \Rightarrow E[S(J_{i-1}, M_{(k+1)_{i-1}})]$

(c) $i = 1, N_i \geq 1, k_i = N_i,$ and $S(J_i, M_{k_i}) = NA \Rightarrow \mu \leftrightarrows \beta$ or $(IB, MB) \leftrightarrows (IE, ME)$ and $E[S(J_1, M_{1_1})]$

(d) $i = 1, N_i \geq 1, k_i = N_i, S(J_i, M_{k_i}) = NA, \mu \not\leftrightarrows \beta$ and $(IB, MB) \not\leftrightarrows (IE, ME) \Rightarrow NI$

(e) $S(J_i, M_{k_i}) = PA \Rightarrow E[S(J_{i+1}, M_{1_{(i+1)}})]$

(f) $S(J_i, M_{k_i}) = TA \Rightarrow NI$

The efficiency of the algorithm is also predicated upon the process that determines the availability of rows and columns and determines the maintenance of the two respective directories. That is, a directory of rows, RM_h, and a directory of columns, RM_v, are maintained where each directory theoretically addresses different sides of the routing board. If required, any subsequent layers or surface mounts are handled similarly.

In general, the determination of the availability at an arbitrary node, (p, l), with respect to either RM_h or RM_v may be addressed accurately, keeping in mind that $H_{p,l}$ and $V_{p,l}$ are interchangeable as they relate to the following explanation (fig. 1). For the sake of specificity, $H_{p,l}$ will be employed here. Thus, let (u_o, l_o) be a target with respect to (p_o, l_o) and where $u_o > p_o$. Then the availability at $(p_o, l_o) = \min[H_{p_o, l_o}, u_o - p_o]$. However, if $u_o < p_o$, then construct the equation $T_o = H_{u_o, l_o} - (p_o - u_o)$. Consider the two cases $T_o \geq 0$ and $T_o < 0$. If $T_o \geq 0$, then the availability at (p_o, l_o) with respect to (u_o, l_o) is $p_o - u_o$. If $T_o < 0$, let $T = H_{u, l_o} - (p_o - u)$, where $u_o < u < p_o$. Then the availability at the node (p_o, l_o) that belongs to RM_h is calculated using the bisection method whose direction of seek is governed by the sign of T. Moreover, because the maximum availability is sought, the number of iterations will be $n = [LOG_2(p_o - u_o)] + 1$.

The integrity of the algorithm, which is implemented by recursive descent, is not disturbed by either the definition of the major and minor axes or the specificity of the terminal and initial nodes. Therefore, all four combinations of axes and nodes are transparent to the algorithm itself. Furthermore, an additional criterion to interchange axes or nodes could be predicated on the value of i or the magnitude of k_i for some i.

The number of vias can be minimized from the contents of the directories $(RM_v$ and $RM_h)$ after the solution is complete. This reduction results in a lower cost for the actual physical implementation of the solution. Finally, the updating of the directories dynamically is trivial, as the RM_v and RM_h contents are corrected based on the values of J_i and the corresponding M_{k_i} resulting from the monotonic sequence $\{J_i\}$.

The application of the theory presented confirms the objectives sought. Appendix A presents examples of typical solutions found. Moreover, a comparison study of the state-of-the-art algorithms is made. Appendix B addresses the efficiency, the cost, the ratio of solution to the Manhattan distance, and the central processing unit time.

CONCLUSION

An automated solution to the printed circuit board problem that circumvents the devastation of exponential growth as a function of complexity has been presented. Moreover, application of the theory displayed in Appendix A demonstrates its versatility and range. The statistics presented in Appendix B confirm the objectives sought. The computer program based upon the ideas presented here will be offered by COSMIC, located in Athens, Georgia.

The author wishes to acknowledge Glenn Bever, Rodney Bogue, and Harry Chiles for their many helpful suggestions.

APPENDIX A
EXAMPLES OF TYPICAL SOLUTIONS*

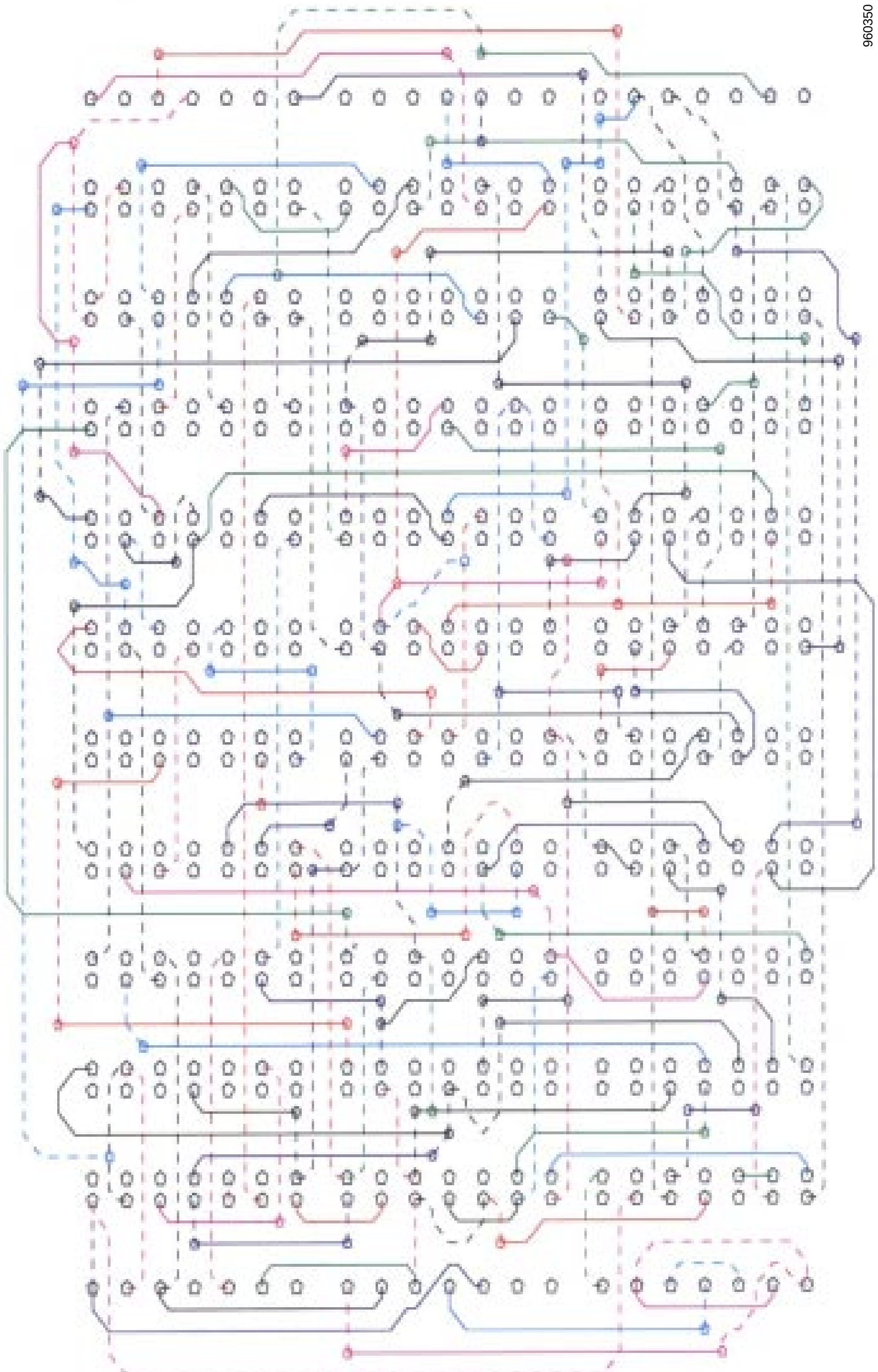

Figure A-1. Board without grid interference considerations; grid size = 0.05 in.

---

*Dash lines and solid lines are on different sides of the layer.

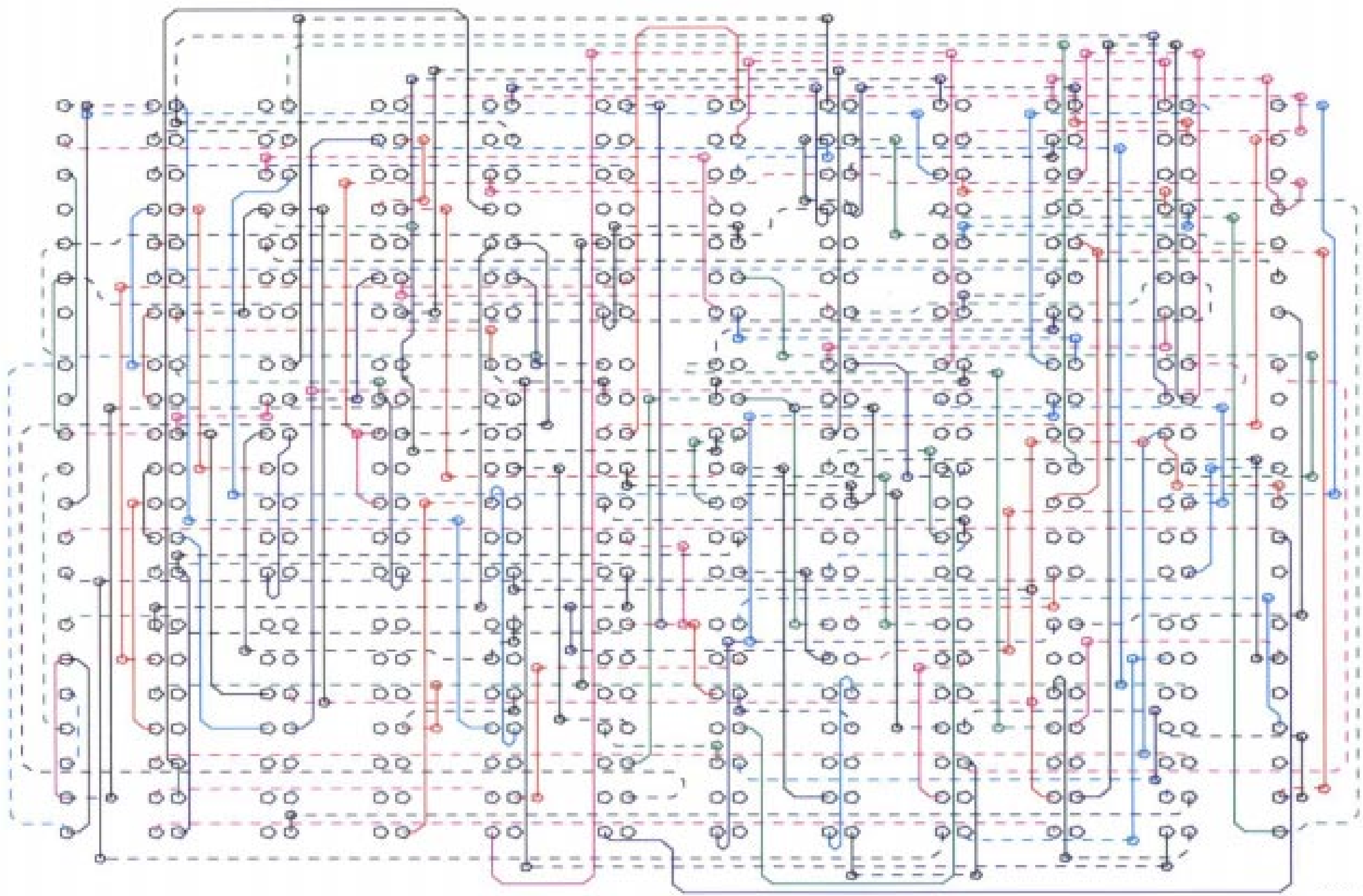

Figure A-2. Board with grid interference considerations; grid size = 0.025 in.

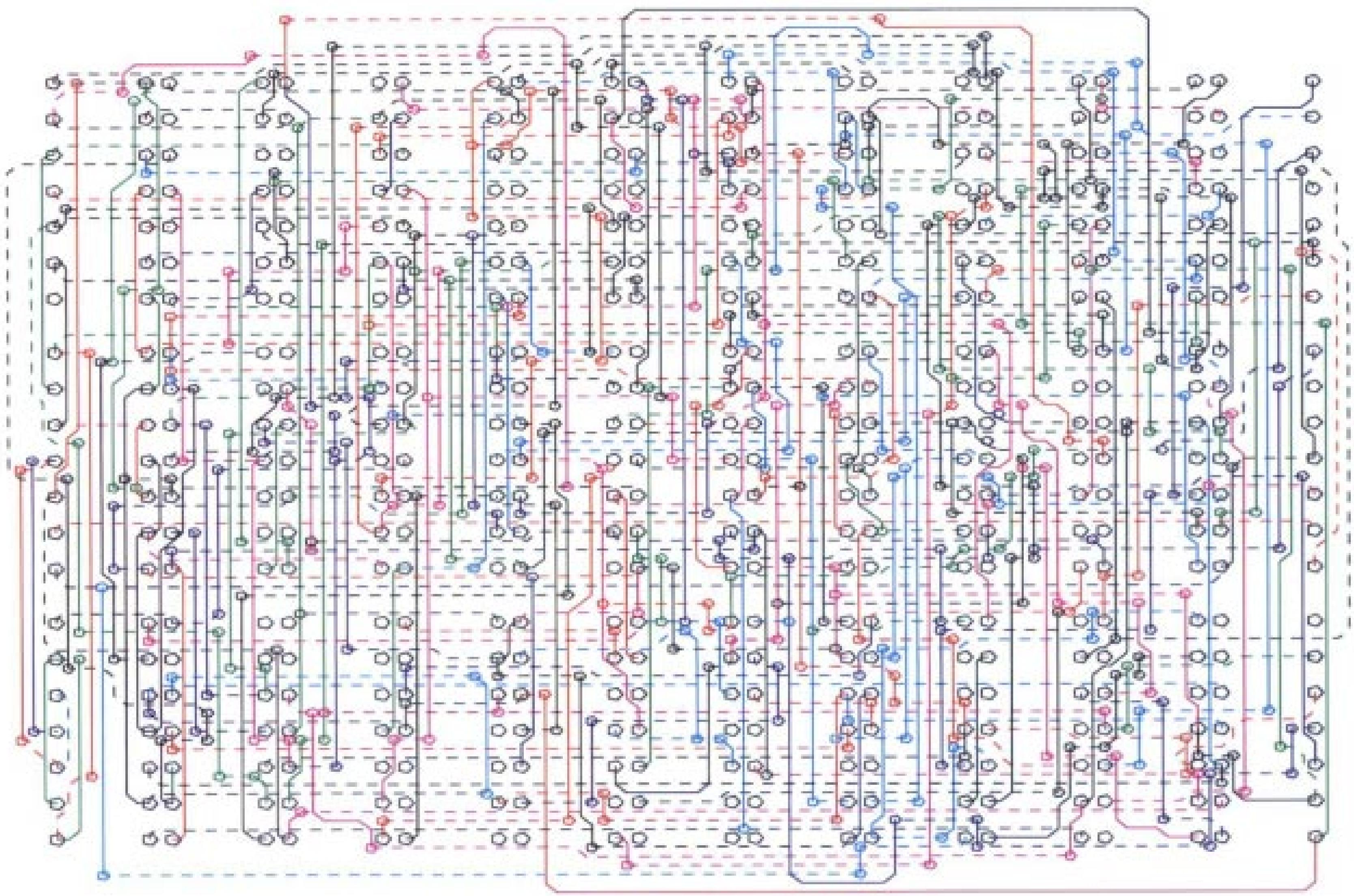

Figure A-3. Board without grid interference considerations; grid size = 0.025 in.

APPENDIX B
TABLE OF COMPARISONS*

Algorithm	Ratio, actual distance/ Manhattan distance	Completion rate, percent	Cost, dollars	CPU time, minutes
Shape tech	1.05	98–100	12,000	10
Pcad	1.10	92	4,000	30
Hedgley	0.97	93	0	0.25

*Chart statistics are reflective of a one-layer board that is two-sided with equivalent grid sizes.

REFERENCES

[1] Kinnucan, Paul, "Printed-Circuit Board Design: Low-Cost Systems are Winning Respect," *The S. Klein Computer Graphics Review*, Summer 1987, pp. 79–105.

[2] Wittenberg, Roland C., "Algorithms Do What Designer Does Naturally," *Electronic Engineering Times*, September 14, 1987, pp. 137–138.

[3] Jackoway, Gary, "Automating the Printed Circuit Board Design Process," *Hewlett-Packard Journal*, February 1988, pp. 68–71.

[4] "Speedy Gridless Software Designs Denser PC Boards," *Electronics*, v. 60, no. 14, July 9, 1987, pp. 102–103.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
September 1996

3. REPORT TYPE AND DATES COVERED
Technical Paper

4. TITLE AND SUBTITLE
A Formal Algorithm for Routing Traces on a Printed Circuit Board

5. FUNDING NUMBERS
WU 505-68-84

6. AUTHOR(S)
David R. Hedgley, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
NASA Dryden Flight Research Center
P.O. Box 273
Edwards, California 93523-0273

8. PERFORMING ORGANIZATION REPORT NUMBER
H-2104

9. SPONSORING/MONOTORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA TP-3639

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified—Unlimited
Subject Category 61

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This paper addresses the classical problem of printed circuit board routing: that is, the problem of automatic routing by a computer other than by brute force that causes the execution time to grow exponentially as a function of the complexity. Most of the present solutions are either inexpensive but not efficient and fast, or efficient and fast but very costly. Many solutions are proprietary, so not much is written or known about the actual algorithms upon which these solutions are based. This paper presents a formal algorithm for routing traces on a printed circuit board. The solution presented is very fast and efficient and for the first time speaks to the question eloquently by way of symbolic statements.

14. SUBJECT TERMS
Artificial intelligence, Printed circuit board, Routing

15. NUMBER OF PAGES
13

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 2000	Technical Memorandum

4. TITLE AND SUBTITLE

User's Guide for Computer Program That Routes Signal Traces

5. FUNDING NUMBERS

WU 529-50-04-T2RR-000

6. AUTHOR(S)

David R. Hedgley, Jr.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

NASA Dryden Flight Research Center
P.O. Box 273
Edwards, California 93523-0273

8. PERFORMING ORGANIZATION REPORT NUMBER

H-2376

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

National Aeronautics and Space Administration
Washington, DC 20546-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

NASA/TM-2000-209036

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified—Unlimited
Subject Category 61

This report is available at http://www.dfrc.nasa.gov/DTRS/

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This disk contains both a FORTRAN computer program and the corresponding user's guide that facilitates both its incorporation into your system and its utility. The computer program represents an efficient algorithm that routes signal traces on layers of a printed circuit with both through-pins and surface mounts. The computer program included is an implementation of the ideas presented in the theoretical paper titled *A Formal Algorithm for Routing Signal Traces on a Printed Circuit Board*, NASA TP-3639 published in 1996. The computer program in the "connects" file can be read with a FORTRAN compiler and readily integrated into software unique to each particular environment where it might be used.

14. SUBJECT TERMS

Artificial intelligence, Printed circuit board, Routing

15. NUMBER OF PAGES

1 CD-Rom

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Unlimited