Lawrence Berkeley National Laboratory
Recent Work

Title

COMPUTER-AIDED DESIGN OF PRINTED-CIRCUIT ARTWORK

Permalink

https://escholarship.org/uc/item/3z82s3bs

Authors

Zane, Ronald
Harrell, Deanna A. Wilber.

Publication Date

1968-04-09

eScholarship.org Powered by the California Digital Library
University of California

UCRL-18172
cy. 2

University of California

Ernest O. Lawrence Radiation Laboratory

COMPUTER-AIDED DESIGN OF PRINTED-CIRCUIT ARTWORK

Ronald Zane and Deanna A. Wilber Harrell

April 9, 1968

TWO-WEEK LOAN COPY

This is a Library Circulating Copy
which may be borrowed for two weeks.
For a personal retention copy, call
Tech. Info. Division, Ext. 5545

RECEIVE
LAWRENCE
RADIATION LABORA
AUG 8 196
LIBRARY AND
DOCUMENTS SECTION

Berkeley, California

UCRL-18172 cy. 3

32

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Chapter for a book*

UCRL-18172
Preprint

UNIVERSITY OF CALIFORNIA

Lawrence Radiation Laboratory
Berkeley, California

AEC Contract No. W-7405-eng-48

COMPUTER-AIDED DESIGN OF PRINTED-CIRCUIT ARTWORK

Ronald Zane and Deanna A. Wilber Harrell

April 9, 1968

*Chapter for Computer-Oriented Circuit Design, Franklin F. Kuo and Waldo S. Magnuson, Jr., Eds. (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1968). (without the Appendices)

COMPUTER-AIDED DESIGN OF PRINTED-CIRCUIT ARTWORK

Ronald Zane and Deanna A. Wilber Harrell

Lawrence Radiation Laboratory
University of California
Berkeley, California

April 9, 1968

ABSTRACT

The problems of designing printed-circuit artwork are discussed. A brief description of manual methods is used to introduce the advantage of automating portions of the design. Emphasis is placed on the interaction between man and machine as the optimum method of production. A program which is based on man-machine interaction is described. It is capable of designing two-sided boards with a variety of components. The appendices include a user's manual describing input procedures and an interpretation of output. A complete listing of the current version of the program is also included.

Chapter 12
COMPUTER-AIDED DESIGN OF PRINTED CIRCUIT ARTWORK

R. Zane, Lawrence Radiation Laboratory, Berkeley
D. A. Harrell, Lawrence Radiation Laboratory, Berkeley

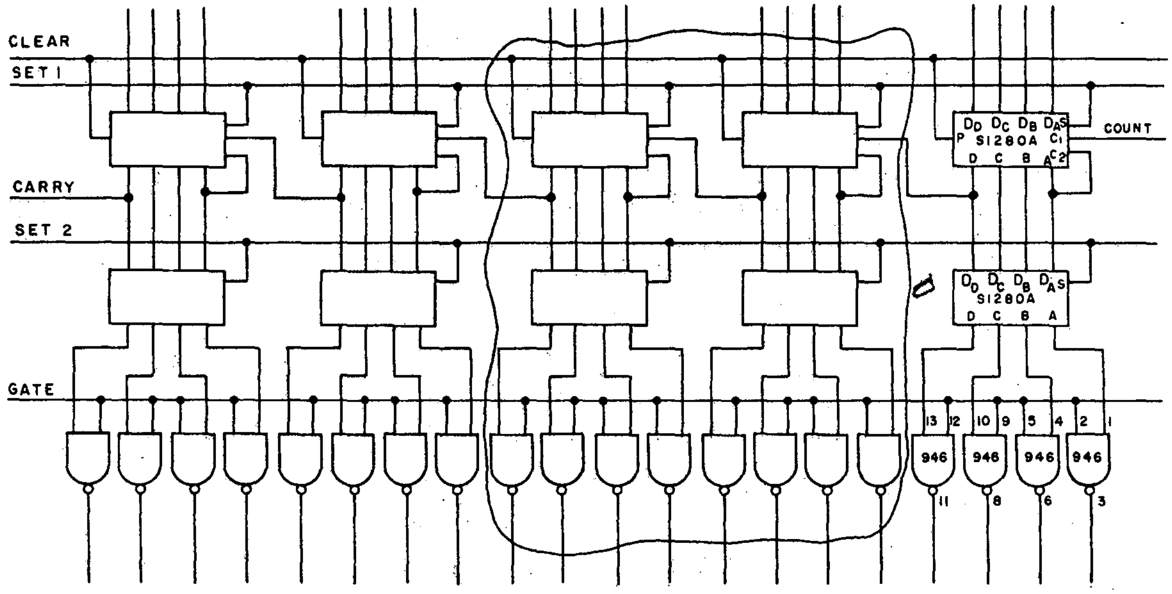
12.1 PRINTED CIRCUIT CARD DEVELOPMENT

The starting point in the design of printed circuit cards is the schematic diagram, logic diagram, or Boolean algebraic expression for the circuit. The circuit is usually a part of a larger system so that a common first step is deciding which portions of the overall system should be confined to a particular printed circuit card. This step establishes the <u>boundaries</u> of a particular circuit, Fig. 12.1.

The printed circuit card <u>board</u> <u>outline</u> chosen for a particular circuit greatly affects the layout of the circuit. The dimensions of the card and the type of connector used are usually dictated by systems packaging considerations, but the effect of the outline on the topology of the circuit must not be neglected. The more important considerations are:

1. Pin spacing and number of pins in the connector.
2. Board dimensions and aspect ratio relative to the connector.
3. Number of surfaces on the board, i.e., single-sided, double-sided, or multi-layer.
4. Power distribution and filtering techniques as well as the presence or absence of the need for a ground plane.

Numerous techniques are available for the etching of printed circuit cards. The choice of technique ranges from the rather crude prototype technique of direct application of an etch-resistant tape or paint to printed circuit laminated stock, through the use of a photoresist such



XBL684-2597

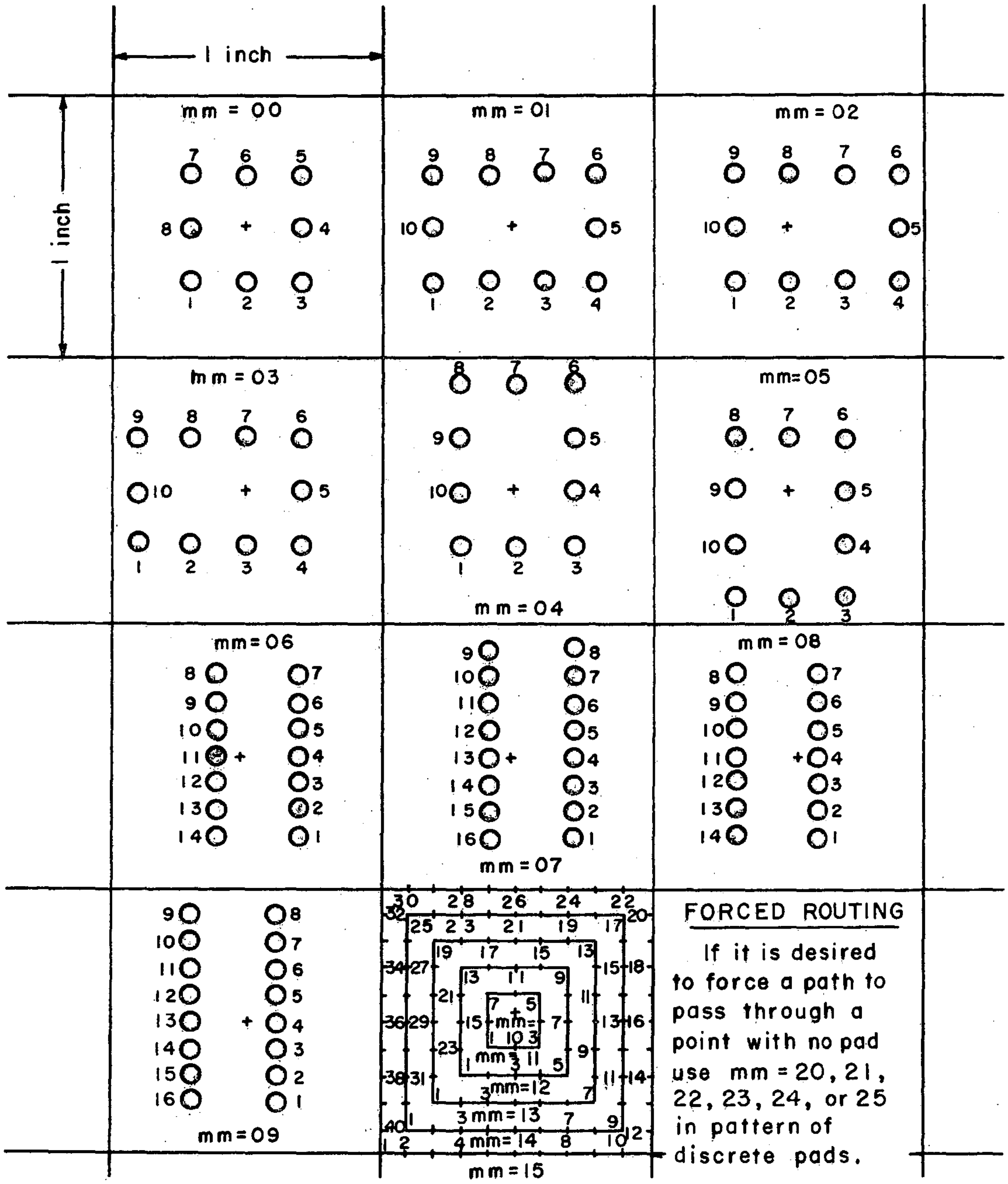
Fig. 12.1. The outlined portion of the Scaler and Register circuit would be allocated to one printed circuit card. All lines crossing the outlined boundary would constitute connections to the printed circuit connector.

as KPR (Kodak Photo Resist) for small production quantities of good quality, to silk screening techniques for large production quantities.

The loading of boards with components is considerably eased if a systematic placement of components is used. To this end it may be desirable to align resistors and capacitors in neat arrays. Integrated circuits may usually be placed in systematic patterns, Fig. 12.2. On the other hand, particularly in high frequency linear circuits, design considerations may preclude the use of systematic placement and instead require the placement of the components in positions and orientations determined by performance characteristics of the functional circuit.

12.2 THE ROLE OF THE COMPUTER

The highly variable nature of the considerations briefly touched upon in the preceding section suggest that the construction of a general algorithm to deal with all the known variables in printed circuit card design is a formidable task. In addition, the construction of such an algorithm is complicated by the fact that technological innovations in circuitry and packaging are prone to introduce new sets of variables in the problem. The complexity of the problem suggests that any attempt to turn it entirely over to a computer program would require either that the program be exceptionally detailed, with many complex involutions and ramifications to its algorithms, or that the variables be handled by generating <u>sets</u> of programs for particular classes of circuits. Either of these approaches would require the expenditure of considerable programming effort. Such large and complex programs would be increasingly difficult to modify to keep pace with the changing state of technology.



FORCED ROUTING

If it is desired to force a path to pass through a point with no pad use mm = 20, 21, 22, 23, 24, or 25 in pattern of discrete pads.

XBL679-5275

Fig. 12.2. PUZZLE has a library of components including 8-pin, 10-pin, 14-pin, and 16-pin packages as well as provisions for discrete points. Standard positioning within a 1-in. square facilitates ready use of the micro-logic.

The answer to the problem of handling the variables in printed circuit card design while keeping pace with the state of the art may lie in programming efforts that rely heavily on man-machine interaction. In such a program the designer would retain control over certain design decisions, whereas the computer would solve topological problems and furnish a high-quality graphic output suitable for printed circuit card production with a minimum number of intervening processes. Ideally the printed circuit card designer would present his circuit to the computer by means of a sketch on a CRT with a light pen, a Rand Tablet, or some form of scanning system. The computer would present a CRT display of its solution for modification by the designer. And the process would be repeated until a solution was produced which the designer could accept. Then the computer would produce a high-quality graphic output by means of a mechanical or optical plotter.

The original motivation for seeking a solution to a problem of this nature may quite properly be simple intellectual curiosity in response to a challenging problem. A continuing effort to utilize a solution must ultimately be justified in terms of effort compared to result. The aim of programming efforts directed toward the design of printed circuit artwork should be the production of printed circuit boards of acceptable quality at low cost. Elegant programs that produce beautiful topological solutions at inordinate expense in terms of programming time or computer time are self-defeating.[1,2,3]

12.3 ROUTING LOGIC.

Once a schematic has been sketched and the necessary components have been chosen, it is time to design the card. There are several methods currently being employed to handle this phase. Some of these are manual and others are automated. Though the procedures required by the manual and automated approaches are dissimilar in many respects, they have some common characteristics. Once a board size is chosen, it is necessary to position the necessary components. This placement will be influenced by such considerations as placing interconnecting components close together, placing components connecting to the connector tabs near the respective edges, spacing components in order to evenly distribute board density, etc. Either approach may permit modification of placement during actual routing of interconnections. However, the time and amount of modification depend on the actual routing philosophy.

In addition, electrical and practical considerations impose certain restrictions on layout. One of the most obvious concerns is to avoid short circuits. Therefore lines that cross each other must appear on opposite sides of the board or be rerouted unless they are part of the same branch of the circuit. Parallel lines on opposite sides of the board, which are superimposed upon one another and separated only by the dielectric material of the printed circuit board, should in general be avoided because they would have a relatively large capacitance between them. It is desireable to minimize overall wire length, in order to minimize the inductance and to optimize use of board space. The number of holes should be minimized in order to reduce production cost. All of these factors must be considered whether the printed circuit artwork is manually or automatically produced.

12.3.1 Manual Design

When a man sets out to design a printed circuit card, he has several distinct advantages over a computer program. The primary advantage is that a human designer may employ a more flexible variety of procedure, whereas a computer is rigidly confined to a predetermined set of rules. A man will first establish a rough layout of the components. His next step would probably be to route the obvious connections, keeping lead length as short as possible and attempting to avoid crossing through the board, yet all the while planning ahead in order to leave area for the other paths. As he proceeds to the more difficult connections he may begin to cheat a bit. For example, he may shift components or related lines. When the most difficult connections are to be made, he may take advantage of the idiosyncracies of the particular circuit. For example, the capacitance effect may be a relatively minor concern for the particular pair of lines involved. Or it may be feasible to extend the leads on a particular resistor in such a way that its terminals are 0.5 in. apart rather than 0.4 in., and consequently allow an additional path to cross the line of the resistor. He is not restricted to a particular grid, hence curved lines and diagonals are readily available to him. In general, he has a free space to work with and the capability to keep an accurate and overall picture, and this visual image can constantly cause him to alter his ground rules.

There are disadvantages to the manual system, however. Two practical concerns are time and cost. A man may spend many hours trying to route a connection, and later have to reroute portions of the path or the entire connection. He may also have to shift large blocks of routing, which is a slow and tedious procedure. This can be costly. Also there is the unavoidable problem of human error. No system can be completely independent of this factor, but under automated systems it can be minimized. The human designer may also tire and resort to jumpers sooner than necessary. The above reasons are some of the factors which are causing the development of automated systems that are faster and less costly.

12.3.2 Automated Systems

Basically two different design philosophies are presently being used in the automated systems for printed circuit design. A three-dimensional model is used by the topological approach. Other systems maintain a two-dimensional model.

The first step in the topological approach is to set up a three-dimensional model with peaks, slopes, plains, and valleys based on the component pin locations.* So again the first step is to locate components. A grid† is then set with weighting factors to represent the board's terrain. These weighting factors may also be influenced by voltage considerations. As paths are routed the topography of the model changes, with new paths creating ridges. The valleys are chosen as preferred paths, and the ridges and peaks are the unavailable area. An incrementing process is then used to work the path through from origin to destination.[2,3]

Interested people have played maze-solving games for centuries. With the advent of the computer, automating chess and similar games has yielded many algorithms to approximate human effort. As the result

*This is the system currently being used by Sandia Corporation and the Thomas Bede Foundation in their system, ACCEL.

†The establishment of a fixed grid is one of the disadvantages of the automated systems. However, because wires cannot be forced to be too close together, if the grid is made fine enough it may very closely approximate the usable working space.

of similar experimentation, the Lee Algorithm[4] for maze solving was developed. The procedure Lee used was to set up two series of concentric wavefronts, radiating from the origin and from the destination within a maze. These wavefronts were set up originally as circles which are forced to yield to obstacles. The path from any point on a circle is then begun by the line segment forming the perpendicular to the tangent of the circle. Line segments are added by succ ssively taking the perpendicular to the tangent of each inner circle at the point of intersection of the previous line segment. The building of wavefronts ceases as soon as the outer wave fronts intersect. Then the line segments are determined. Lee's Algorithm can assure a solution if one exists and can guarantee the shortest path. The implementation of Lee's Algorithm per se requires an unreasonably long computer run, therefore modifications have been made by concerns using this approach.* However, such systems may still be expensive in terms of computer cost.

At Lawrence Radiation Laboratory we are using a simplified stepping procedure. In some ways it resembles both the topological and the Lee approach; however, its basic precepts are markedly different. The logic very closely resembles that of a rat in a maze. The only knowledge our hypothetical rat has is the general direction in which he must go. Motion that reduces the distance to the destination is defined as positive, and motion that increases this distance is defined as negative.†

*Modified forms of Lee's Algorithm are being used by Fridén, San Leandro, and by Norden Division of United Aircraft.

†At the present time the only motion allowed is orthogonal, i.e., along grid lines. At a future time diagonals will be incorporated allowing eight possible direction choices rather than four.

The rat may move in only a horizontal or a vertical direction, and as he completes each new path, it is blocked to prevent his reuse of it. The rat must be assumed to have tunnel vision, that is, he can see only in the direction he is moving. When he runs into a wall, he backs away from it and turns 90 degrees and starts in that direction. He then moves until he hits another wall, then rotates again, and so on until he reaches his destination. He has been trained sufficiently that if he is blocked when he changes direction, he goes back to his original direction, backs up one more space, and again tries to turn. One additional trick has been installed in the rat's maze. That is, his maze is a two-level maze, with all vertical paths in one level and all horizontal paths in the second level. (In the real situation these two levels correspond to the two sides of a board.) Therefore, in order to change direction the rat must change levels. It is when this level change is impossible that the rat must move backwards.

The rat has been given a bit of a helping hand. When he works himself into an impossible situation, he is put at the beginning again and given a new starting direction. He is given a total of four tries to start out. If the fourth try fails, his destination and starting point are reversed and he is given four more tries.

The sophistication necessary in the routing techniques will vary with the individual needs of an electronic production organization. The greatest advantages seem to be gained, however, by keeping the logic for the machine simple and employing some form of man-machine interaction to achieve the more sophisticated logic.

12.4 IMPLEMENTATION OF PUZZLE

The problem of computerizing various phases of printed circuit card development has been handled in several different ways.[1,2,3,4,5,6,7] Some of these methods involve highly sophisticated algorithms and consequently require several hours of machine time. These are the programs based on the terrain methods or involved maze-solving techniques discussed in the previous section. Other systems depend on some type of man-machine interaction and hence allot the user more control over the finished board.[1,4,5,6,7] These systems have the added advantage of minimizing the costly computer time necessary for the completion of a circuit card. Such a system is used at Lawrence Radiation Laboratory.

The basis of LRL's system is a program called PUZZLE.[5,6,7] PUZZLE is currently running on a CDC 6600, using a core of approximately 125000 (octal), and spending approximately 10 to 15 seconds of execution time.*

12.4.1 Basic Assumptions

Before any actual coding can be done for a system, a working framework must be established. The ground rules for PUZZLE were determined after considerable discussion and experimentation. These rules centered around such things as component placement, board size and dimensions, line spacing, and scaling factors for the graphic output.

Most of the rules came from considerations outside the computer itself. For example, what size boards were actually needed? What components were available and currently being used? If graphs were

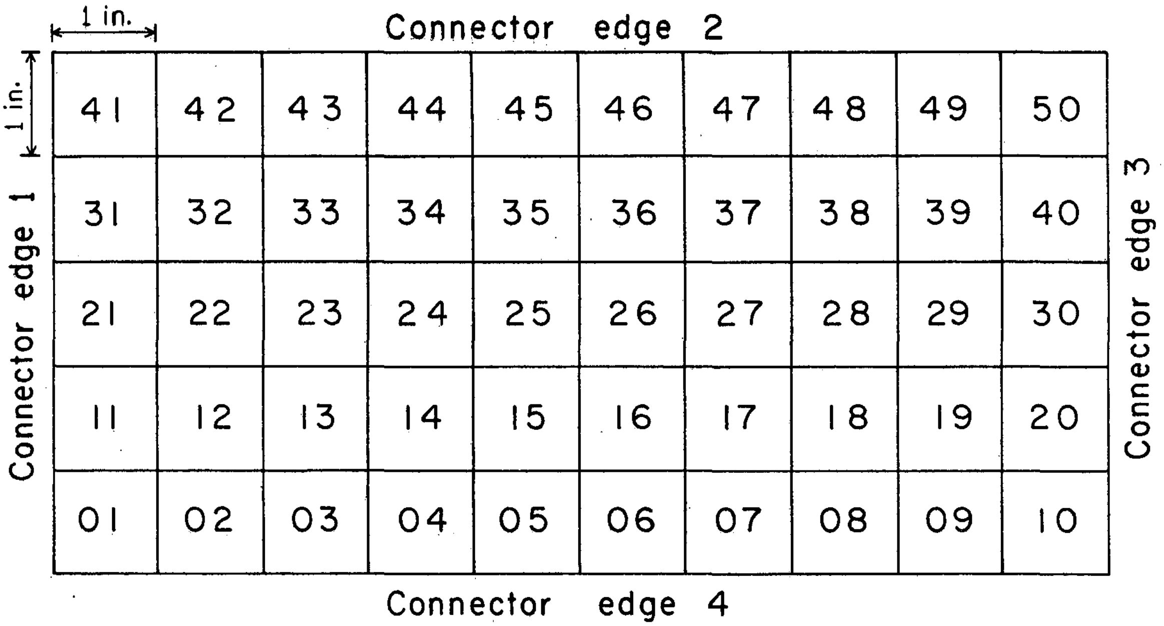
*The current Software system at LRL is CDC's Chippewa. PUZZLE is written entirely in Fortran IV and it would be reasonably convenient to convert the program for use on other machines.

to be drawn twice scale, what effect would photographic reduction have on pad size and line width? What line width would be necessary with current fabrication techniques to insure reliable circuitry? In addition there are limitations imposed by both hardware and software. These include the size of the plotting surface,* the grid nature of the pen motion on the CalComp plotter, pen widths available with the plotter, and increments used in plotting. Choices were also influenced by programming difficulty.

It was decided that it would be convenient to use 0.1 in. spacing for lines and pads, because component leads and pins are usually fabricated for use on 0.1-in. centers. A double-sided board would be produced with essentially all horizontals on one side and all verticals on the other.† In addition, to simplify input, the working area is blocked into 1-in. squares. The present version of PUZZLE can handle up to fifty of these squares in a 5×10 array, Fig. 12.3; hence the working area maximum is 5×10 in. Any rectangle within this array may be used. Also the circular configuration for eight- and ten-pin micro-logic presented a problem. Therefore it was decided that in order to fit the grid pattern, it would be convenient and quite workable to use a square configuration for the eight-pin and a rectangular configuration for the ten-pin micro-logic, Fig. 12.2. Then a centered position was chosen for these packages with rotations provided for the ten-pin can. At a later date the dual-in-line packages were incorporated, allowing two positions for each. The

*LRL is using a 565 CalComp drum plotter. The drum is 11 in. wide.

†This has been modified within the logic of the program, but all plotting is begun with this assumption.



41	42	43	44	45	46	47	48	49	50
31	32	33	34	35	36	37	38	39	40
21	22	23	24	25	26	27	28	29	30
11	12	13	14	15	16	17	18	19	20
01	02	03	04	05	06	07	08	09	10

XBL684-2598

Fig. 12.3. PUZZLE's orientation is based on 5 in. × 10 in. rectangle divided into fifty 1-inch squares.

choice of standard positioning is a convenience for the user, not a restriction. Facility is also provided for specifying discrete points within the grid area. This facility allows the user to use variable-length resistors, inductors, etc., to place transistors, and to shift the larger packages in other than the standard position. These various restrictions were then used as the framework to build a working system.

12.4.2 Data Transmission

A unique coordinate system was devised to facilitate easy definition of the points to be connected. Each point of the board is specified by a combination of three codes: the type of component, the square the component falls in, and the pin being specified with the component. The points that are to be interconnected are listed together on a single data card. The program will accept two to eleven of these points per card. There may be as many of these cards as are necessary to describe the circuit. There are two additional blocks of information which may be necessary to complete a board. The first of these is optional. It is a list of variables describing the board dimensions and location of connector tabs. The other is an identifier which includes such information as the user's name, date, job number, and the scale to be used for the graphic output. PUZZLE has been designed to recognize any such identifiers as the beginning of a new data group. This allows the user to sequentially run several boards through the computer in a single run.

With some working experience we found that unnecessary failures were occurring because data points were improperly coded. Consequently, provisions have been added to PUZZLE to examine the data as thoroughly as possible and delete any data that cannot be interpreted. A message is then printed to indicate to the user what data has been deleted. The remainder of the data is processed normally.

12.4.3 Board Outline

Earlier versions of PUZZLE yielded graphic output which called for some manual tape laying. That is, the user was responsible for adjusting the layout which PUZZLE produced within an appropriate board outline and then using a tape overlay to bring connections to the connector tabs. However, the current version of PUZZLE allows the user to code his board dimensions (with variable spacing on the connector tabs). Hence PUZZLE now yields a complete representation of the finished board. Any or all edges may have connector tabs on them, and within the physical limits any number of connectors may appear on the tab. It is also possible to produce a board with no connector tabs.

If the specifications for the board outline are omitted or found faulty or incomplete, the outline will be omitted and the program will revert to the previously used method. That is, a scan is made of the data to determine the maximum and minimum in each direction. A working area is then described which includes the necessary maxima and minima. The area will be $\underline{x}$ by $\underline{y}$ inches, where $\underline{x}$ and $\underline{y}$ are integers. Hence during the data scan it is necessary to check for the square inch a component is located in, not the actual physical coordinates of the point.

12.4.4 Sequencing

Very little effort has been made in the routing logic to optimize path length. It is evident that in general by decreasing path length, the remaining working area is increased, and hence more interconnections may be completed. Therefore two levels of data sequencing have been introduced to indirectly reduce path length.

The first step is to order the individual data strings. These are often not listed in the most convenient order. Therefore a procedure is used to obtain the shortest total path length for the string. The string is broken into pairs and no restriction is made in their configuration other than optimizing length. That is if points *A*, *B*, *C*, and *D* are to be connected they might all be in a string *A* to *B*, *B* to *C*, and *C* to *D*, but if the total path length would be shorter they could be paired in such a way that *B*, *C*, and *D* are each connected to *A*.

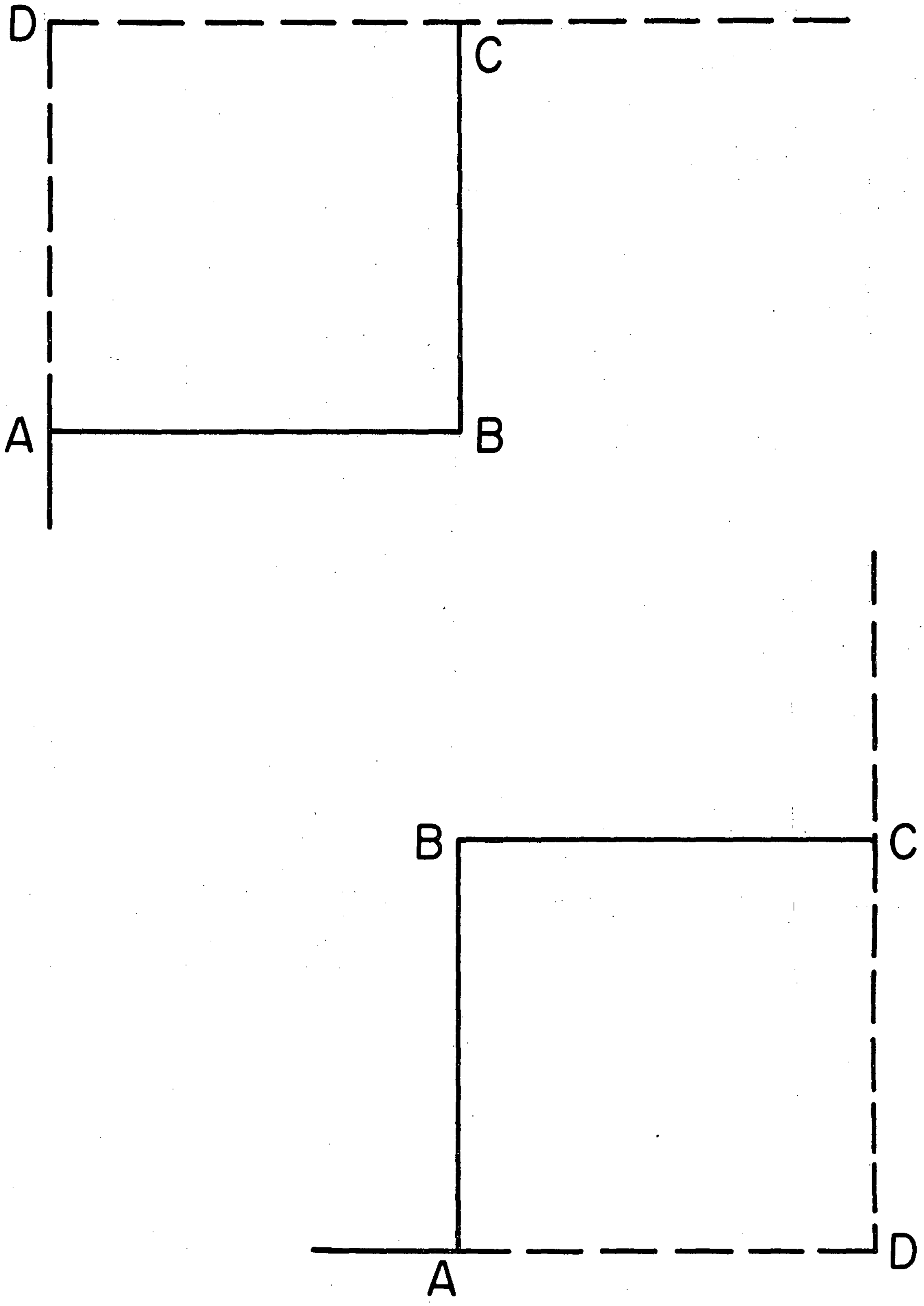
A system of matrix operations is used to perform the necessary ordering. One matrix, **QM**, is used to hold the path length for any possible pair. (Because path length does not depend on direction, only the triangular matrix is necessary.) A second matrix, **BM**, is used to record what connections are yielded as choices are made among the possible pairs. This Boolean matrix is initialized with a truth value of false.

The first step is to fill the first matrix, **QM**, with the indicated path lengths. Because of the grid restrictions that routing move either horizontally or vertically rather than diagonally, path length is based on the rectilinear length and any obstacles are ignored. The matrix is then scanned for the shorest path length. The appropriate data pair is chosen as the first leg of the branch, and its path length is set at a maximum so it will not be selected again. The appropriate location in the second matrix is assigned a truth value of true. A second pair of points is then chosen and the appropriate value is set in BM. To obtain the full extent of the connections now yielded, it is necessary to square BM. (Using the Boolean "and" operator is equivalent to algebraic multiplication.) For example, if AB were the first choice and AC the second, squaring BM fills the location corresponding to BC. The choosing process continues until BM has a full truth value of true. At each step BM receives one new value of true and is operated on until the new value of $BM=(BM)^{N}$, where N is the number of choices that have been made.

The second level of sequencing concerns the entire data block. It is based on the premise that by routing a long and involved path prior to short one, unnecessary obstacles may completely block the simpler path. In order to prevent this problem, after the data has been broken into connecting pairs, the data is scanned and ordered so that the shortest paths will be routed first.

12.4.5 Routing in PUZZLE

The routing procedure in PUZZLE depends on successful interrelation between several somewhat simple phases. The basic mapping is an implementation of the rat-in-the-maze technique described in Section 12.3.2. Initially a positive vertical and positive horizontal direction are chosen. These are the orientations necessary to yield the most direct path to the end point. Hence if A is going to B, and A is to the left and above B, right and down are the positive directions. The path is then begun using increments in the positive vertical direction. This path continues either until it reaches an obstacle, such as a predefined path or a pad necessary for the placement of a component, or until it reaches the vertical coordinate of the endpoint. This basic pattern is followed until a connection is completed. Of course only the simpler routes will be completed by the direct stepping process. Therefore there are various provisions for backing up on a line, extending a line beyond the endpoint and dropping back into the destination, restarting a line, or causing the line to clear a component. Arbitrary limits have been set to prevent a line from backing up too far (such as beyond the origin), or extending the line too far beyond the end point. A line may be restarted in several ways. A total of eight tries is possible. The first try calls for starting in the positive vertical direction, the next in the positive horizontal, the third in the negative vertical, and the fourth in a negative horizontal direction. The next four tries parallel these, except that the origin and the destination are reversed. Because of the greater difficulty encountered when a path passes through a component, various segments cause a line to avoid moving into the center of a component and being blocked. As a path is being traced it may develop unnecessary kinks or bends; hence a *smoothing process* has been built into the routing algorithm. Each time the program causes the path to change direction, the two previous line segments are examined. If it is possible, the line segments are redetermined in order to remove a bend. In Figure 12.4, if ABC comprise the two completed line segments and a direction change is desired at C, hen if no obstacles block the path ADC, ADC would eliminate at least one and possibly two bends in the overall path. (If A or C is a terminal, no bend would appear at that point, therefore the smoothing process would eliminate only one bend.) At first glance this smoothing



XBL687-2999

Fig. 12.4. There are two possible inputs to the smoothing procedure, i.e., at the initiation of a vertical line segment or at the origin of a horizontal line segment. In both cases ADC yields a smoother final path than does ABC.

seems unnecessary due to the stepping procedures. Direction changes usually result when an obstacle is encountered and hence the smoothing process would achieve nothing. However, the facilities for retracing a line, for moving around a unit of obstacles, for going beyond the end point, etc., make the smoothing necessary.

Each of these principles has been used to increase the number of completed paths. However, there may be some connections that resist all efforts to be routed. Should such problems occur the program searches for other connections to correctly complete the circuit. That is, if A, B, and C are to be connected and in that order and B cannot be connected to C, the program attempts to connect A to C in order to complete the circuit.

12.4.6 Graphing Procedure

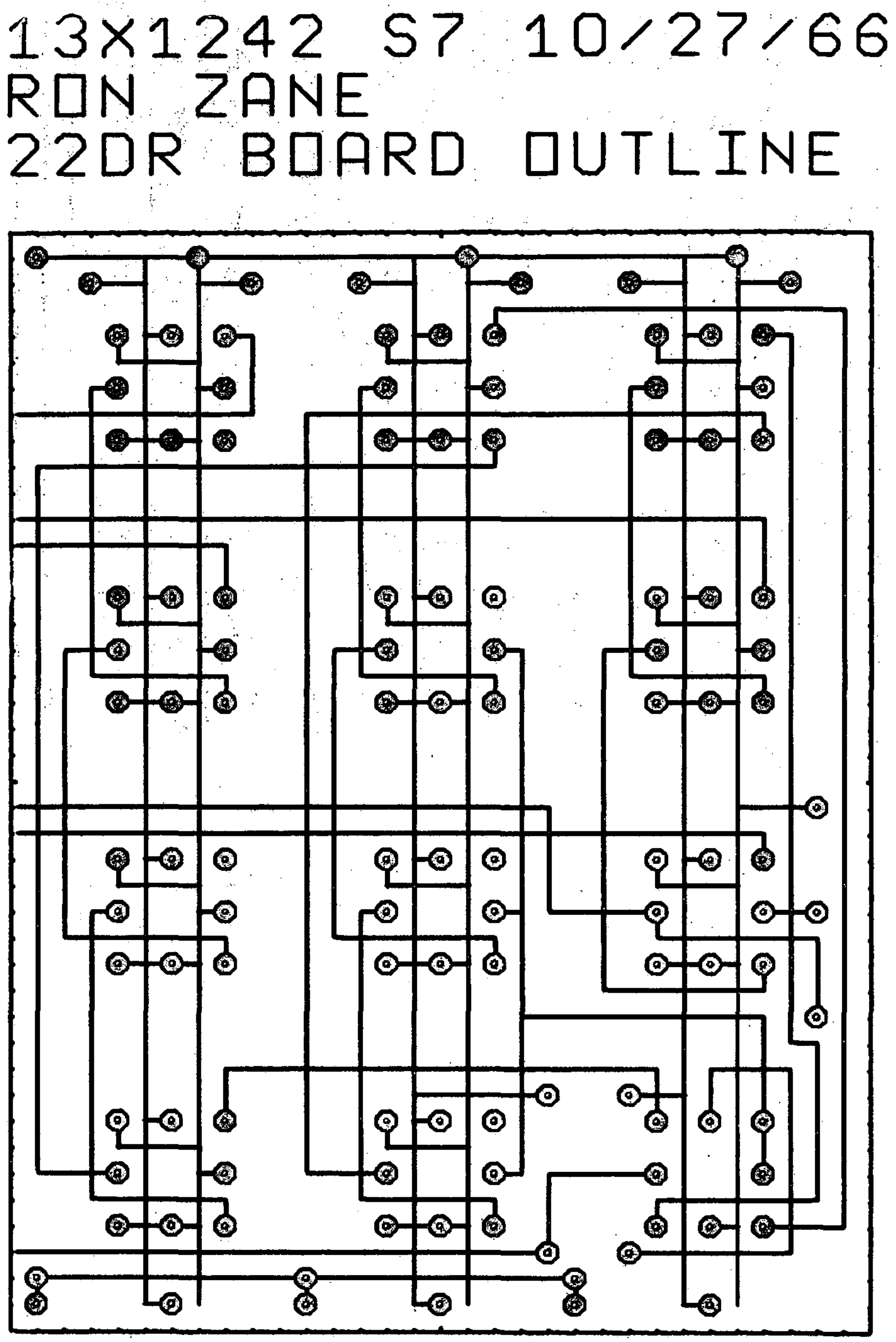
The graphing procedure, which consists of two portions, begins with the mapping of a composite picture of both sides of the board. This graph is sufficient for debugging purposes and becomes a convenient reference for board loading. The composite graph will always be drawn.

The second phase of the graphing is optional, as it consists of graphing the individual sides of the board, and it is a waste of both computer and plotter time to plot these graphs on a run which does not complete all paths. Hence, when a graph is incomplete due to either bad data or missing connections, PUZZLE branches around this section.* It is at the beginning of this stage that the various line segments are assigned to a particular side of the board. A complete path

*There is an exception to this rule. Provisions have been added to PUZZLE to force the graphing if the user so chooses.

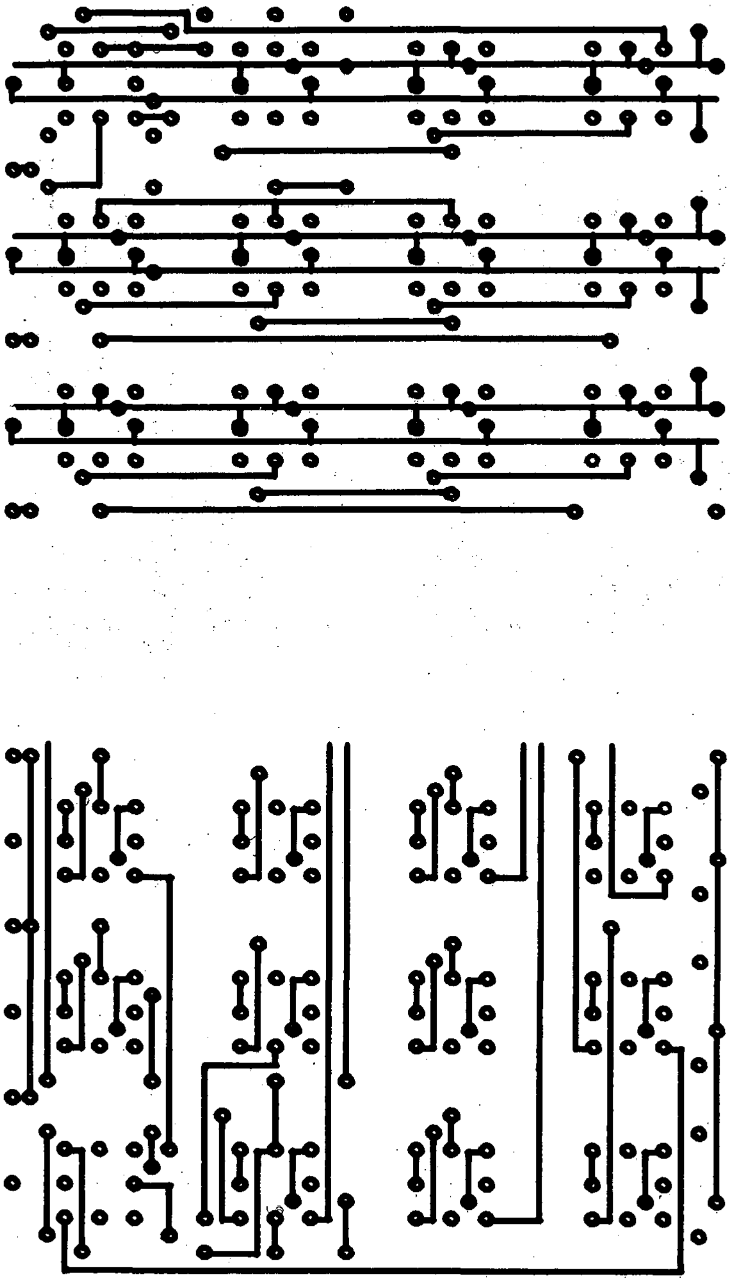
is examined and any line segment that crosses a line perpendicular to it is assigned first, all vertical lines being assigned to the first drawing and all horizontals to the second. Any remaining segments are then assigned in such a way as to avoid unnecessary holes through the board. Should an entire path be unrestricted, it will appear on the second graph. The second graph is a mirror image, thus allowing the user to place the two graphs back to back to duplicate the composite.

Figures 12.5 through 12.8 show the composite graph, graphs of each side of the board, the artwork with tape overlay to the connector (the border outline option was not used), and the finished board. A simplified flow chart of PUZZLE is shown in Figure 12.9.



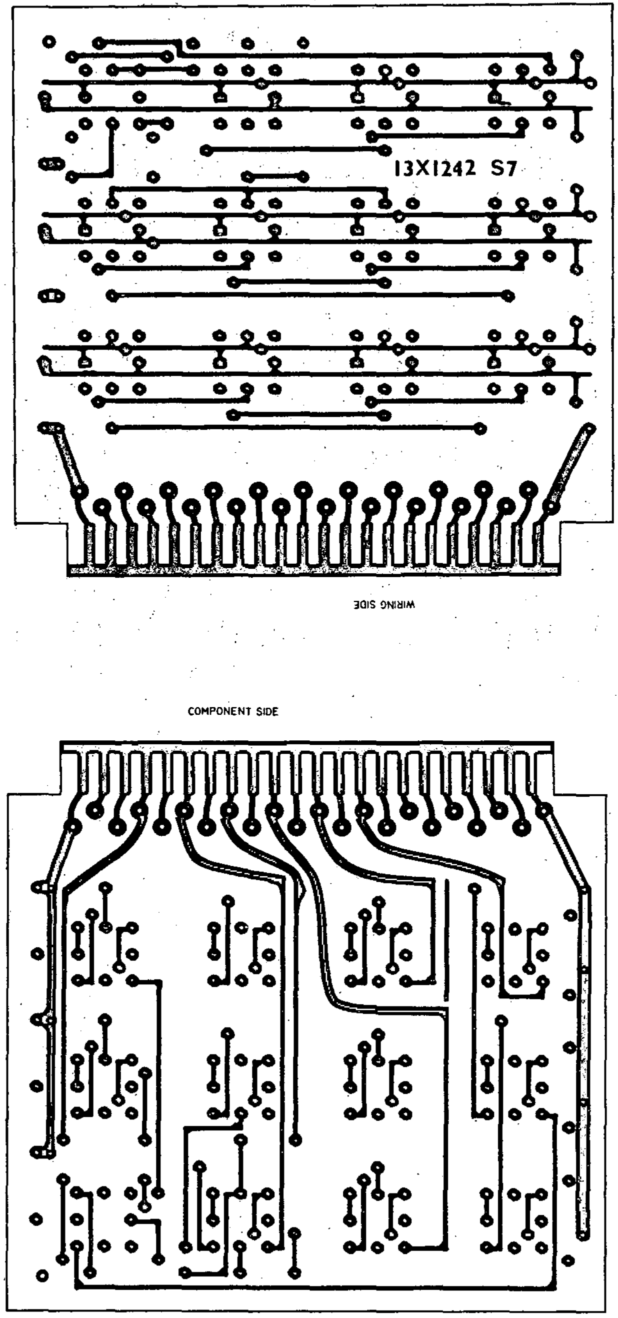
XBL 679-4935

Fig. 12.5. The first graph drawn by PUZZLE is a composite picture of both sides of the board as viewed from the parts side.



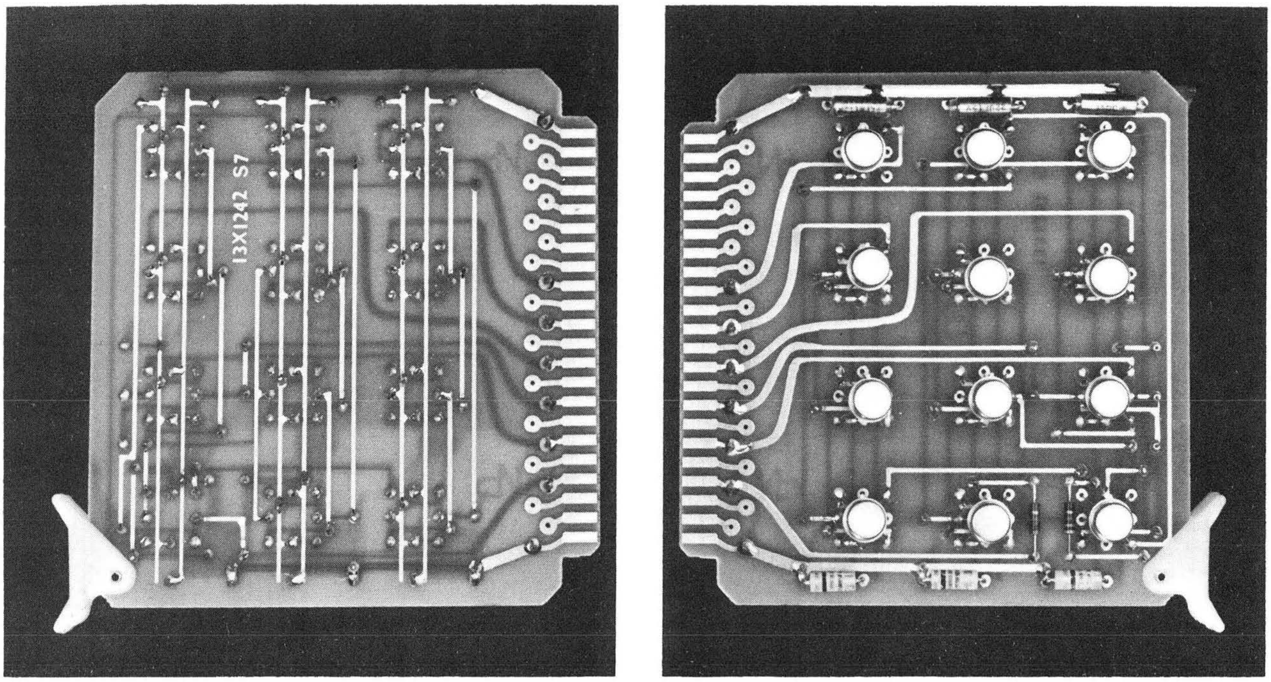
XBL 679-4936

Fig. 12.6. The second and third graphs are pictures of the parts and wiring sides.



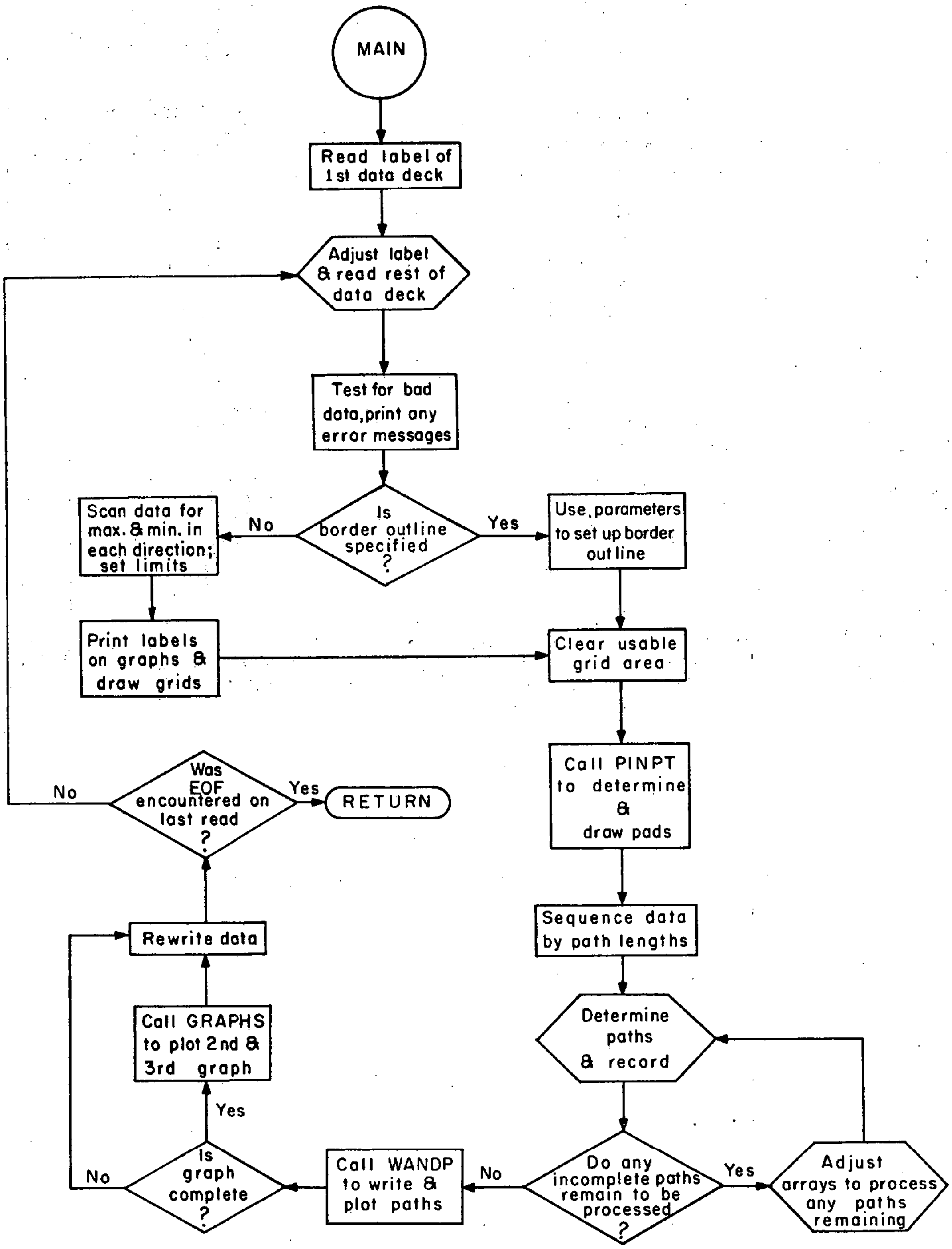
XBL 679-4934

Fig. 12.7. When no border outline is specified, it is necessary to place the graphs produced by the CalComp into a board outline and lay tape to the connector.



Elec. 4238

Fig. 12.8. The finished board.


MAIN
Read label of 1st data deck
Adjust label & read rest of data deck
Test for bad data, print any error messages
Is border outline specified ?
No
Yes
Scan data for max. & min. in each direction; set limits
Use parameters to set up border outline
Print labels on graphs & draw grids
Clear usable grid area
Call PINPT to determine & draw pads
Sequence data by path lengths
Determine paths & record
Do any incomplete paths remain to be processed ?
Yes
Adjust arrays to process any paths remaining
No
Call WANDP to write & plot paths
Is graph complete ?
Yes
No
Call GRAPHS to plot 2nd & 3rd graph
Rewrite data
Was EOF encountered on last read ?
Yes
RETURN
No

XBL684-2596

Fig. 12.9. The flow chart for PUZZLE.

12.5 LIMITATIONS AND EXTENSIONS OF PUZZLE

PUZZLE has been designed to provide a solution rather than the optimum solution to the printed circuit card design problem. As a result there are several limitations and inefficiencies in PUZZLE's solutions:

1. The path lengths are not minimized. The routing logic tends to produce short paths but no effort is made to assure that they are the shortest paths available.
2. Through holes are not minimized. The smoothing routine tends to reduce the number of holes carrying lines back and forth through the surface of the boards but the algorithm does not minimize them.
3. Solutions to portions of the topological routing may be incomplete, requiring unnecessary jumpers or touch-up of the artwork.

In line with the basic philosophy of the design of PUZZLE, projected improvements in its performance are predicated upon efforts to improve its solutions rather than to try to optimize them. Some of the modifications that will possibly be incorporated in the near future are:

1. Enable the program to utilize the 45-degree diagonal paths. Geometrical and topological considerations indicate that this procedure will insure adequate line separation while increasing the number of available path routings by about 40%.
2. Require that the program test all available route solutions

and use the shortest one. The present logic stops seeking paths as soon as a trial completes the path.

3. Relocate the lines which block a path when all attempts to complete a given path have met with failure. The offending lines could be eradicated to permit completion of the blocked path and then the eradicated line could be retried to find an alternative path.

The CalComp pen and ink plotters have proven themselves to be of limited value in the production of the high quality graphic output needed in the production of printed circuit artwork. The pen and ink drawings may be used to produce prototype boards by direct reproduction, but if the high quality needed for a production run is to be produced an optical plotting system is indicated.*

*The Lawrence Radiation Laboratory, Livermore, is using a Gerber optical plotting system to plot PUZZLE output. It produces very high quality film positives directly.

APPENDIX 1

PUZZLE IV Input Procedure

PUZZLE's input has been designed to give the user as much flexibility as possible. The data deck may have enough information to do several boards or only enough for one. The number of interconnections to be completed may vary from 1 to 550.[1] And the number of interconnected points in a single branch may vary from two to eleven. The user may consider the information necessary to yield one board a data block. The deck which contains all data blocks for a given run will be referred to as the run deck.

The data blocks will be comprised of three types of cards.

1. The first card of the data block may be considered the label card.

(a) Columns 1 through 60 must contain 60 characters to be used as an identifier on both printed and graphic output. It will appear as a single line in the printed output and as three 20-character lines on the graphic output. (See Appendix 3.) This label may include such information as user's name, 3U number, a date, or the particular application. At least one nonnumeric character must be included in the label.

(b) Column 61 must contain the scaling factor for the graphic output. At present actual-size or twice-scale drawings are the only sizes permitted. The program then expects a 1 or 2 in column 61. If any other character appears, actual-size drawings will be produced and the printout will flag the defective data.

[1] It is possible to increase the number of allowable interconnections by a very minor program modification.

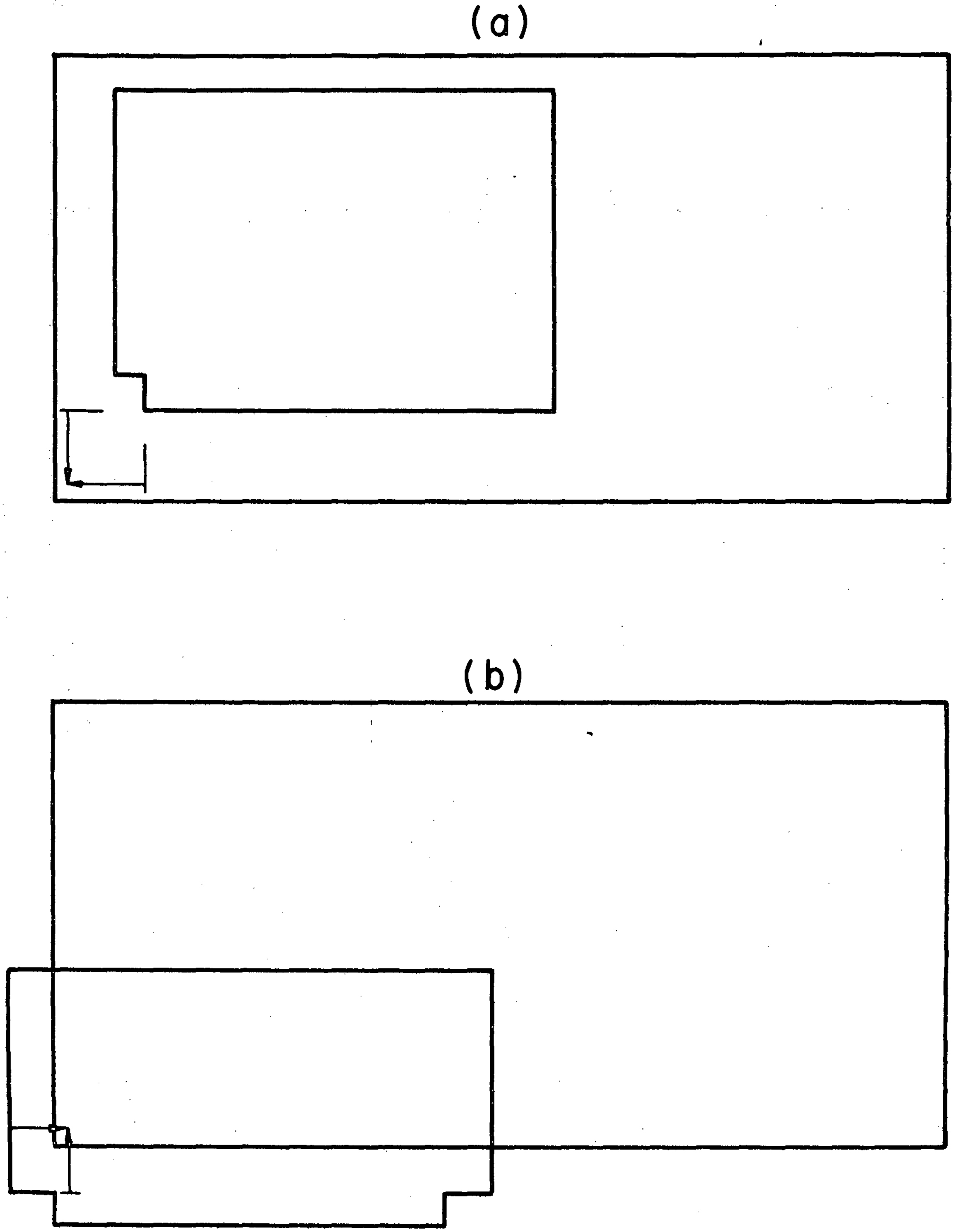
(c) Column 62 is used to control the graphing of the individual sides of the board. A zero or blank will allow the program to control the graphing. That is, if there are any bad data or any missing connections only the composite will be graphed. If the graph is complete, however, all three graphs will be produced. If column 62 contains a 1 all three graphs will be produced regardless of the completeness of the mapping.

(d) Two additional arguments are necessary if and only if the user chooses to use PUZZLE's option of placing the interconnecting components in an appropriate outline which will include connector tabs. Components are placed within a 5 × 10-inch rectangle. (See section on seven-digit codes.) Columns 64 through 70 should contain the horizontal distance in inches from the left-hand lower corner of the board to point (.1,.1), the first usable point in the rectangle. Columns 71 through 77 should contain the vertical distance. If the board is located inside the 5 × 10-inch rectangle these distances will be negative. Figure 10 shows pictorial representations of both the positive and the negative condition.

2. The second data card is optional. If used it should carry the parameters necessary to describe the board outline. If any value was placed in columns 64 through 77 the program will expect this second card to exist.

(a) Columns 1 through 7 should contain the width of the board in inches. The number may be placed anywhere in the seven columns, but a decimal point is required to convey magnitude. Examples of acceptable value are 3.5, 5.0, 7., or .96.

(b) Columns 8 through 14 should contain the height of the board in inches. The acceptable forms of the number are the same as for the width.



XBL687-2996

Fig. 10. a. The distances necessary to locate the board with respect to the 5 × 10-in. grid are negative.
b. The distances are positive here.

(c) Columns 15 through 21 should specify the number of connectors for edge 1 (see Fig. 2). The number must be right-adjusted within the seven columns.

(d) Columns 22 through 28 should specify the vertical distance from the lower left-hand corner. In Fig. 11 this is the value represented by A1. The number should be in inches and the format is the same as that used for the height and width.

(e) Columns 29 through 35 should contain the number of connectors on edge 2.

(f) Columns 36 through 42 should specify the horizontal distance in inches from the upper left-hand corner to the edge of the connector tab on edge 2.

(g) Columns 43 through 49 should specify the number of connectors on edge 3.

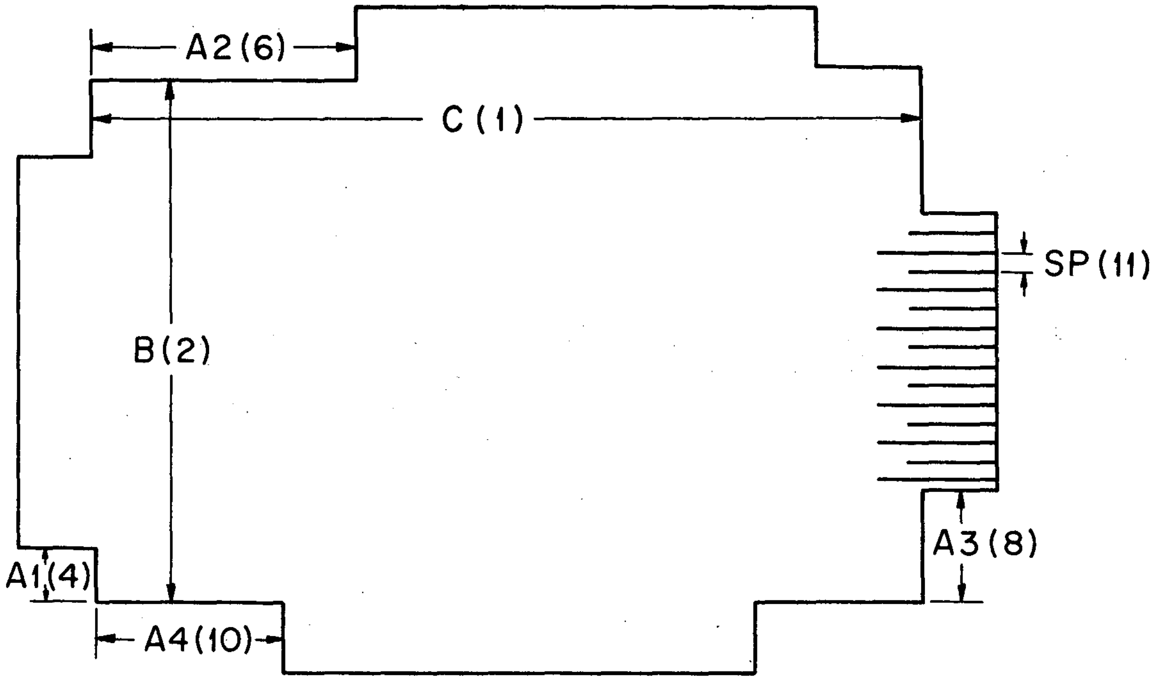
(h) Columns 50 through 56 should specify the vertical distance in inches from the lower right-hand corner to the connector on edge 3.

(i) Columns 57 through 63 should specify the number of connectors on edge 4.

(j) Columns 64 through 70 should specify the horizontal distance in inches from the lower right-hand corner to the connector on edge 4.

(k) Columns 71 through 77 should contain a number to convey the spacing on the connector tabs. This number should be in inches and of the same format as the width, height, and distance measures.

3. The remainder of the data deck constitutes the wiring list. Each individual card will contain a string of interconnected points. The program will continue to look for these cards until it encounters either a



XBL687-2997

Fig. 11. Seven of the eleven parameters needed to specify the board outline are shown here. The number in parenthesis shows each parameter's position on the data card.

new label card or a blank card. Each card may contain up to eleven points, each point being represented by a seven-digit code. The composition of these seven-digit codes is as follows: 0mmnnpp.

(a) The first digit is always zero or blank.

(b) mm is determined in the following manner:

(i) If the unit is an eight-pin micro-logic can,

mm = 00 (or two blanks),

the can will be centered in the 1-inch square (see Fig. 3).

(ii) If the unit is a ten-pin micro-logic can,

mm = 01 (or a blank and a 1)

centers the can in the square (see Fig. 3);

mm = 02

shifts the can 0.1 in. to the right of center (see Fig. 3);

mm = 03

shifts the can 0.1 in. to the left of center;

mm = 04

sets the can in the vertical position with pins 4 and 10 on the vertical center;

mm = 05

places the can in the vertical position with pins 5 and 9 on the vertical center;

mm = 06

places a fourteen pin dual-in-line with the left row 0.1 in. left of center;

mm = 07

places a sixteen pin dual-in-line with the left row 0.1 in. left of center;

mm = 08

places a fourteen pin dual-in-line with the left row 0.2 in. left of center;

mm = 09

places a sixteen pin dual-in-line with the left row 0.2 in. left of center.

(iii) If the unit is a discrete component we have

mm = 1x

where x specifies the distance in tenths of an inch from the center of the square (see Fig. 3).

(iv) If the point is being used for forced routing and a pad is not desired at this point,

mm = 2x,

where again x is the distance in tenths of an inch from the center.

(v) If the point being specified is on the connector edge,

mm = 3x

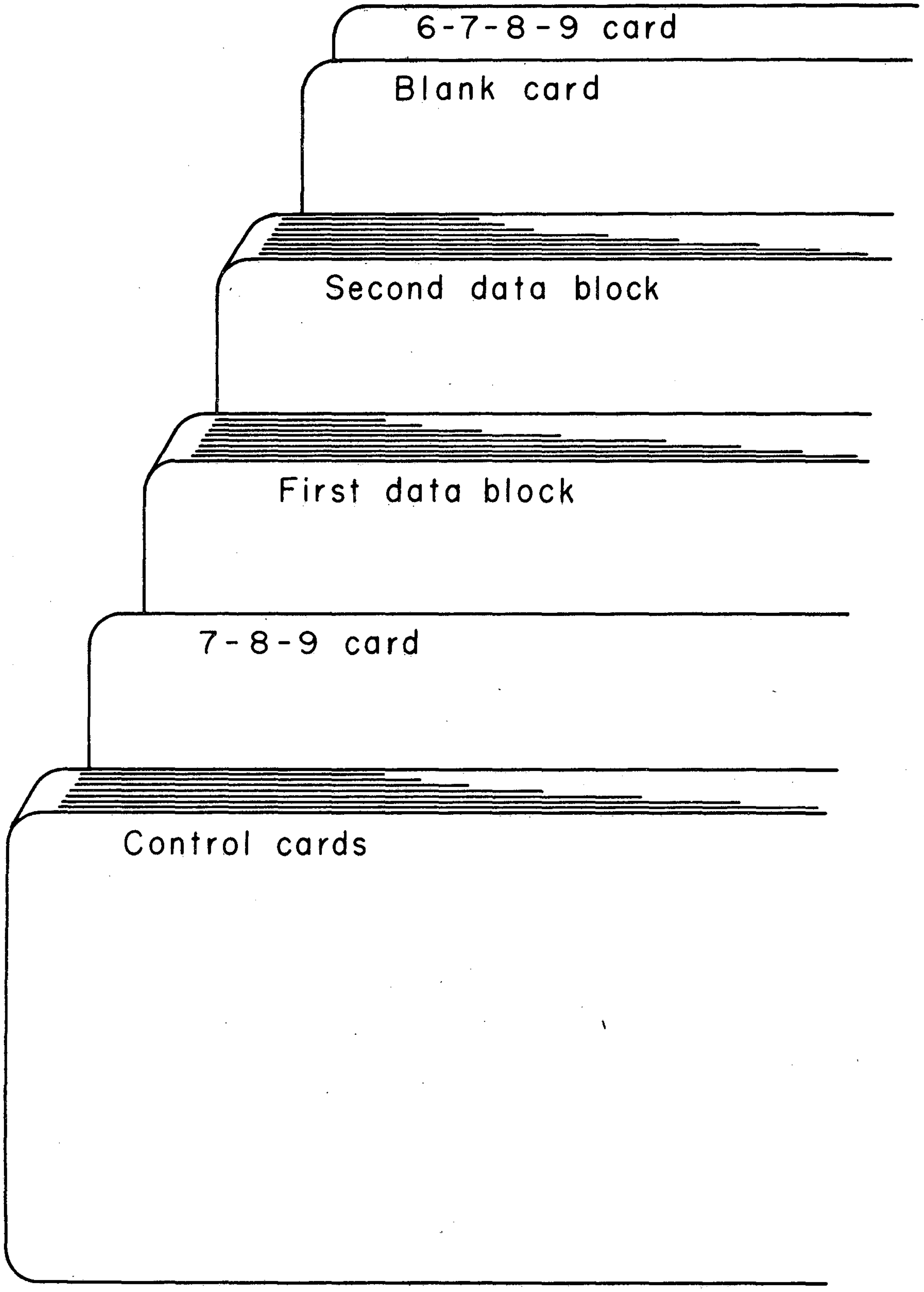
where x is the edge referred to.

(c) The next two digits (nn) refer to the square in which the unit is placed. Figure 2 shows a 5 × 10-in. rectangle. The user may choose to use any portion of this by assigning nn appropriately.

(d) The last two digits are the pin number. Figure 3 shows the location of the pins of the assigned components. For example, to specify the upper left-hand corner of an eight-pin can, pp = 07. For the discrete components pp is the point in the ring (see Fig. 3). If the point is on the connector edge pp equals the connector number. The connectors are numbered from right to left if the connector is facing down (see Fig. 11).

For examples of data construction see Appendix 3.

The actual construction of the run deck includes a control deck and the data blocks necessary for the given run (see Fig. 12). The control deck consists of the following cards, all starting in column 1:



XBL687-2998

Fig. 12. Run deck construction.

```
PUZZLE, 7, 100, 125000. 363201, D. WILBER
REQUEST TAPES. LIBRARY TAPE 194
REWIND (TAPES)
COPYBF (TAPES, MAIN)
UNLOAD (TAPES)
RETURN (TAPES)
REWIND (MAIN)
REQUEST TAPE99. CALCOMP PLOT TAPE
MAIN.
7-8-9
```

The first of these cards has several variables. The first seven columns are for the job name. It may be 2 to 7 characters long with a comma immediately following. The second field is priority. This is fixed. The third field is the time limit. With experience the user can readily adjust this. In general 100 octal seconds[1] is sufficient for about two and a half average-density boards. The fourth field, field length, is fixed. The first field to the right of the period is the user's account number; this is followed by his name and any appropriate comments. The 7-8-9 card is a card with a 7, 8, and 9 punch in column one.

Following the 7-8-9 card are the data blocks. Following the data blocks is a blank card, then a 6-7-8-9 card. The 6-7-8-9 cards are available in the computer ready room. (They have fluorescent edges to aid the computer operators.)

[2] 100 octal = 64 decimal.

APPENDIX 2

Interpretation of Output

There are two forms of output produced by PUZZLE, graphic and printed. Each serves a valuable function.

Printed Output

There are several parts to the printed output. The data will be listed twice, once at the beginning and once at the end of the listing. The first listing is produced as soon as the data are read; it will therefore list any bad data. The final listing will be the list of data actually operated on.

The second portion of the printed output is a listing of any appropriate error messages. These may warn of errors in the board outline, in the scale factor, or in the interconnection list. If the bad data appear in the interconnection list, that one datum will be deleted and the remainder of the deck will be processed normally. When this occurs an additional line will be printed (item 7) giving the user an approximate location of the bad datum within the data deck. The error messages include:

1. BAD DATA-FIRST COLUMN OF SEVEN-DIGIT CODE NOT ZERO.
2. BAD DATA-COMPONENT CODE IS NOT VALID.
3. BAD DATA-PIN CODE DOES NOT AGREE WITH COMPONENT CODE.
4. BAD DATA-CONNECTION CALLED FROM NONEXISTENT SQUARE.
5. BAD DATA-CONNECTOR CANNOT BE MAPPED FOR THIS BOARD.

6. BAD DATA-THIS CONNECTOR NUMBER GREATER THAN ALLOWED FOR THIS BOARD.

7. xxxxxxx WAS DELETED FROM THE DATA LIST, IT IS THE xxxx PIECE OF DATA AND IS THE xxx ITEM ON THE CARD.

8. MISSING DATA-SIZE SPECIFICATION FOR GRAPHS 2 AND 3 IS MISSING OR IS INCORRECT. IT WAS SET TO ONE.

9. AN ERROR WAS FOUND IN YOUR BOARD SPECIFICATIONS-OUTLINE CANNOT BE DRAWN.

10. AT LEAST ONE DATA POINT EXCEEDS THE BOARD DIMENSIONS-BOARD DIMENSIONS WILL BE IGNORED.

The third portion of the printed output gives the user the results of PUZZLE's processing. The majority of this will be a listing of completed routing: PATH FOR PIN PAIR xxxxxxx yyyyyyy WAS COMPLETED. At the end of this listing will be a message, PATHS COULD NOT BE FOUND FOR THE FOLLOWING INTERCONNECTIONS, followed by the codes for any missing connections.

Graphical Output

The most important part of the output produced by PUZZLE is the representations of the board produced by the CalComp plotter. There are three graphs produced when the routing is completely successful. If there are any incomplete paths, only the first or composite picture will be mapped. The first graph pictures the board from the parts side but shows both sides. This composite picture is very useful for checking for bad placement of components or for bad data unobserved by the program's error check, and for the general density of the board or of given segments of the board, and as a guide to the actual builder of the board as to placement of components. If the program determines that for any reason this board is incomplete, that is, if there are any incomplete paths or rejected data, this composite picture is the only one produced, and may be used to adjust the input data to correct the insufficiencies.

The second graph shows all lines which are to appear on the "parts side" of the board. These will be predominantly vertical lines. The third graph shows the "wiring side" of the board, and will be essentially all of the horizontal lines.

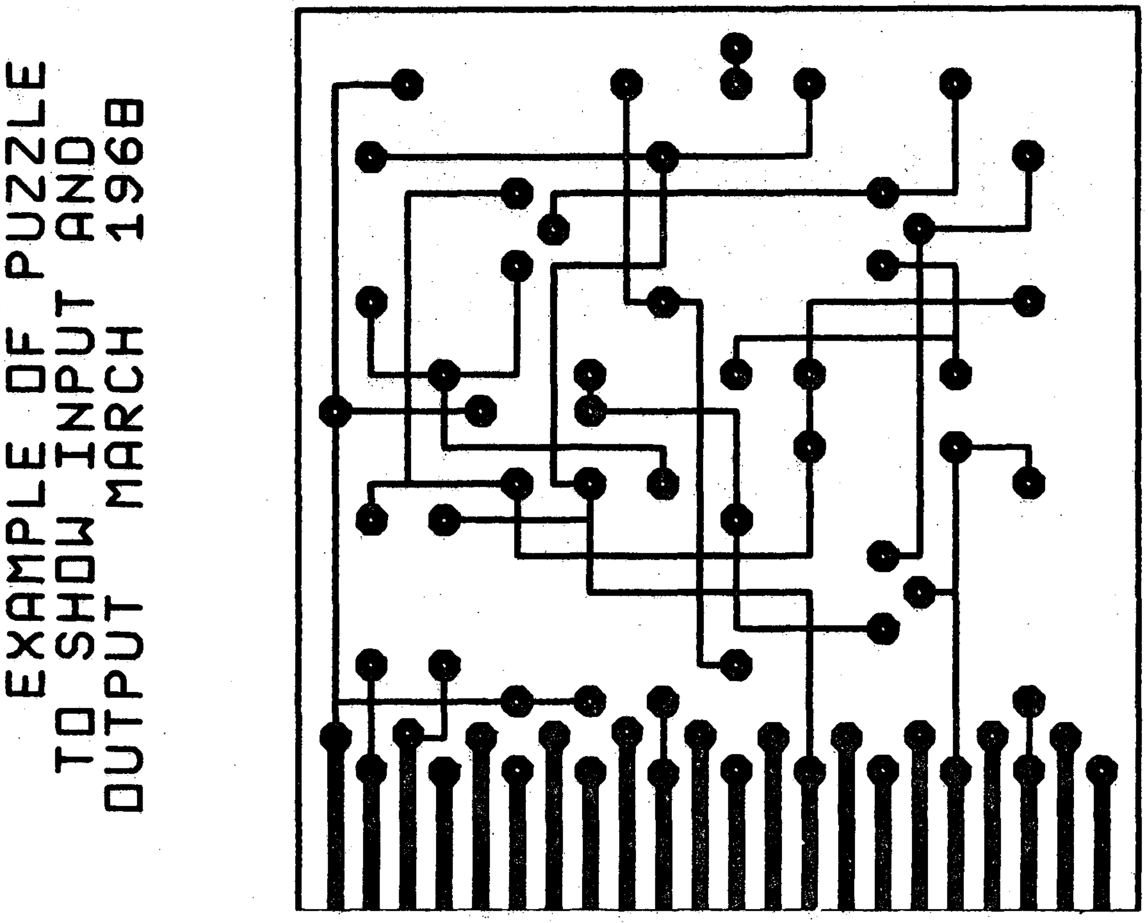
Experimentation was done to determine the best materials to use for this graphical output. The standard CalComp paper is thin and not so stable as is desired for best results. Therefore the final graphs are plotted on vellum, which is produced by the makers of the CalComp. This is much more stable than the paper, and the lines are sharper. In addition, a pen tip wider than that normally used on the plotter is used to produce lines wide enough for direct production of the board from the CalComp graphs. These should be requested by the user when the job is submitted.

APPENDIX 3

Figures 13 through 18 show results of complete and correct input data. Figures 13 through 15 show an example using the board outline on connector edge 4. (Note that the connectors are staggered.) The label has been adjusted to break conveniently into three 20-character lines. Figures 16 through 18 show the same board without the board outline. Note that the wiring is very similar, differing only for those connections involving the connector.

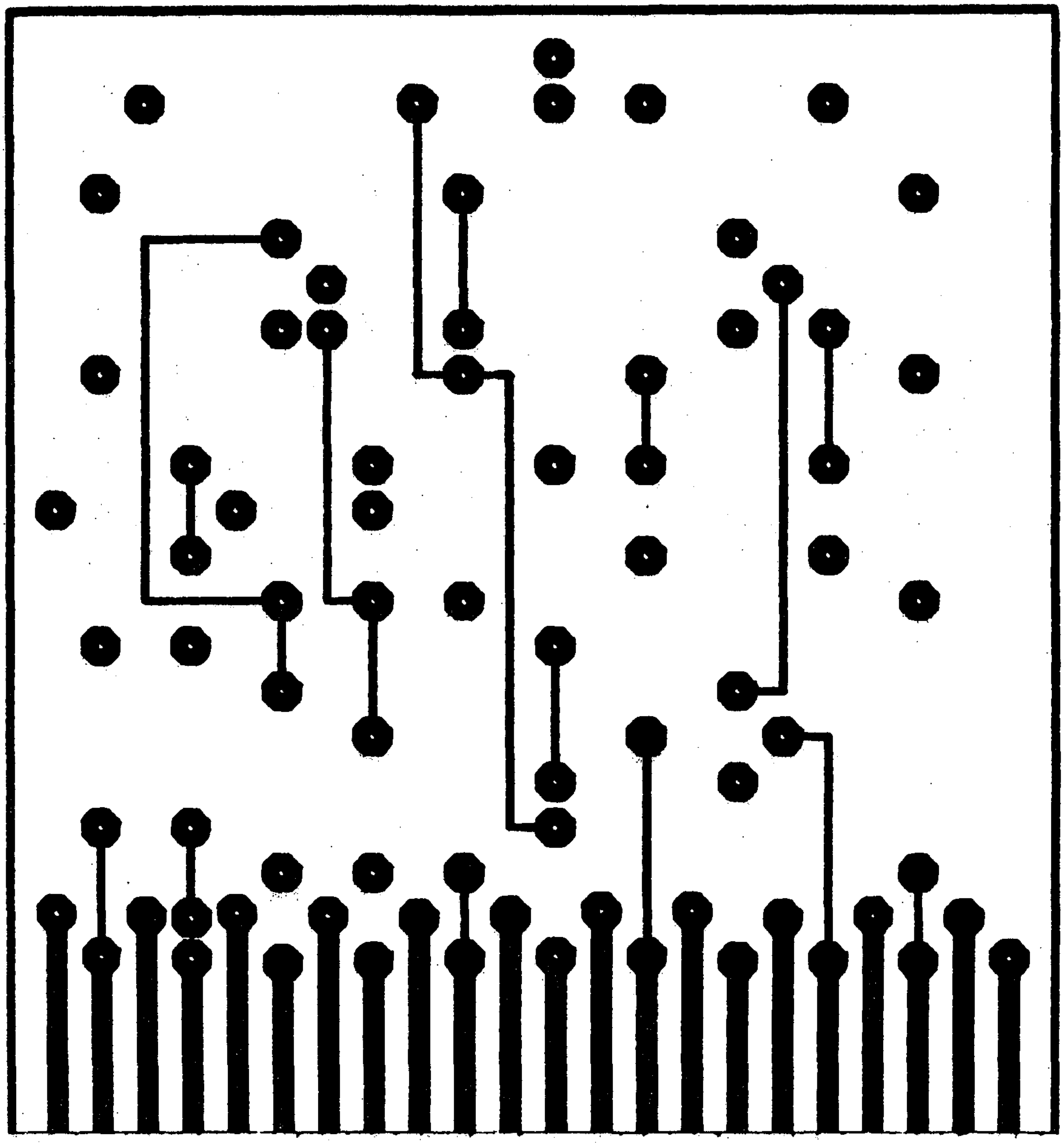
Table I shows a listing of the punched input deck for a board with board outline. Table IV shows the data for the same board without the connector edge. Note the difference in data coding in the second, ninth, tenth, eleventh, and twelfth cards.

Tables II, III, V, and VI show the various parts of the printout produced when the graphs are made. The data which the machine reads is listed, a summary of routing success is given, and a list of the actual processed data is included.



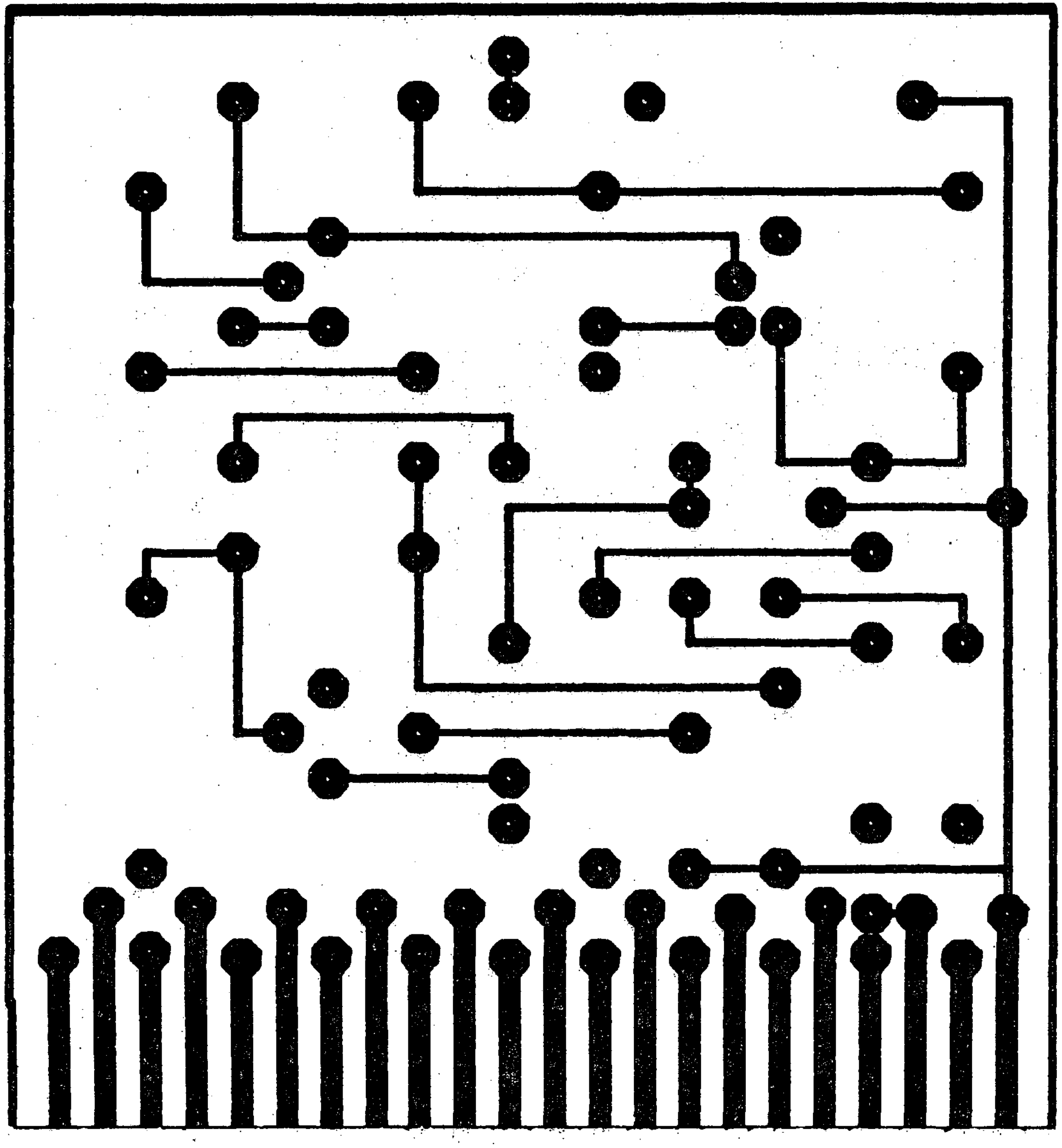
XBL 686-1138

Fig. 13. Example of PUZZLE output, a composite showing board outline and project label.



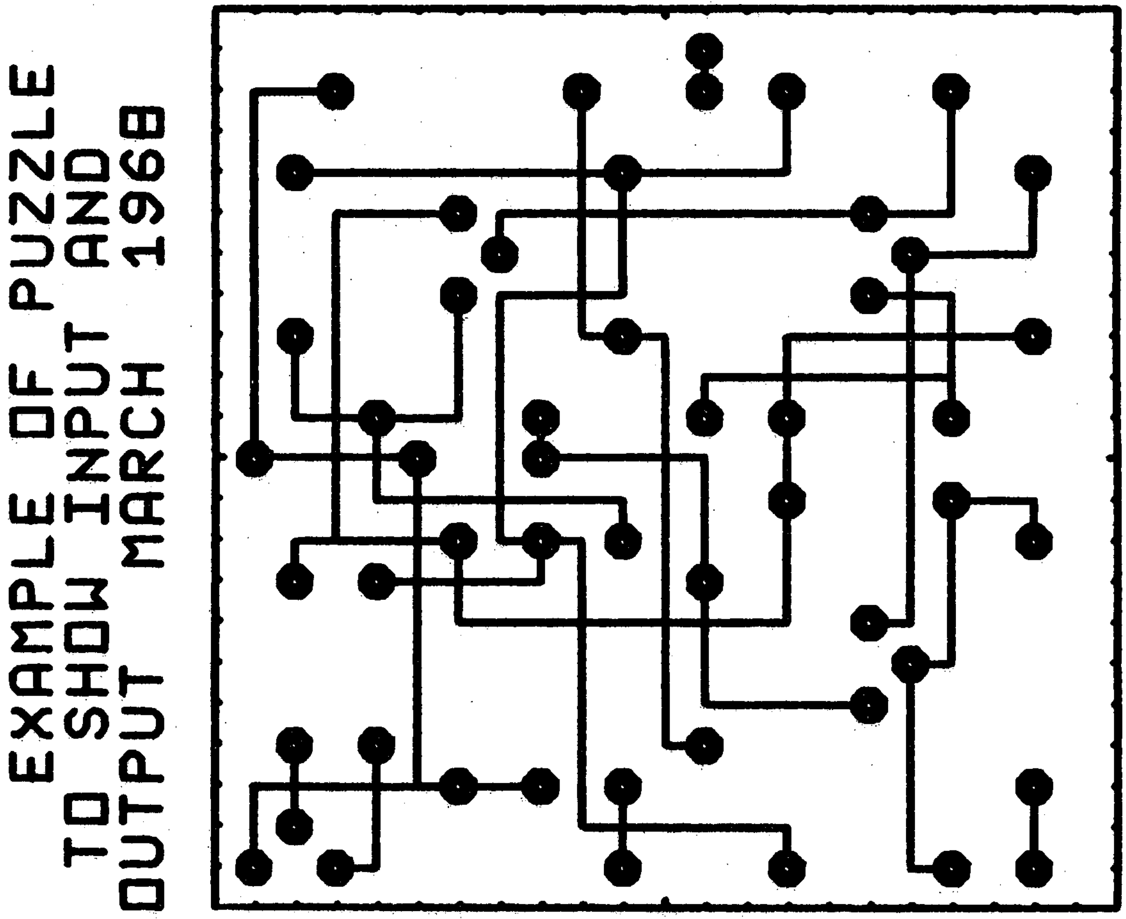
XBL 686-1139

Fig. 14. A graph of the parts side (connector 1 is on the right). Note that most of the wires are vertical.



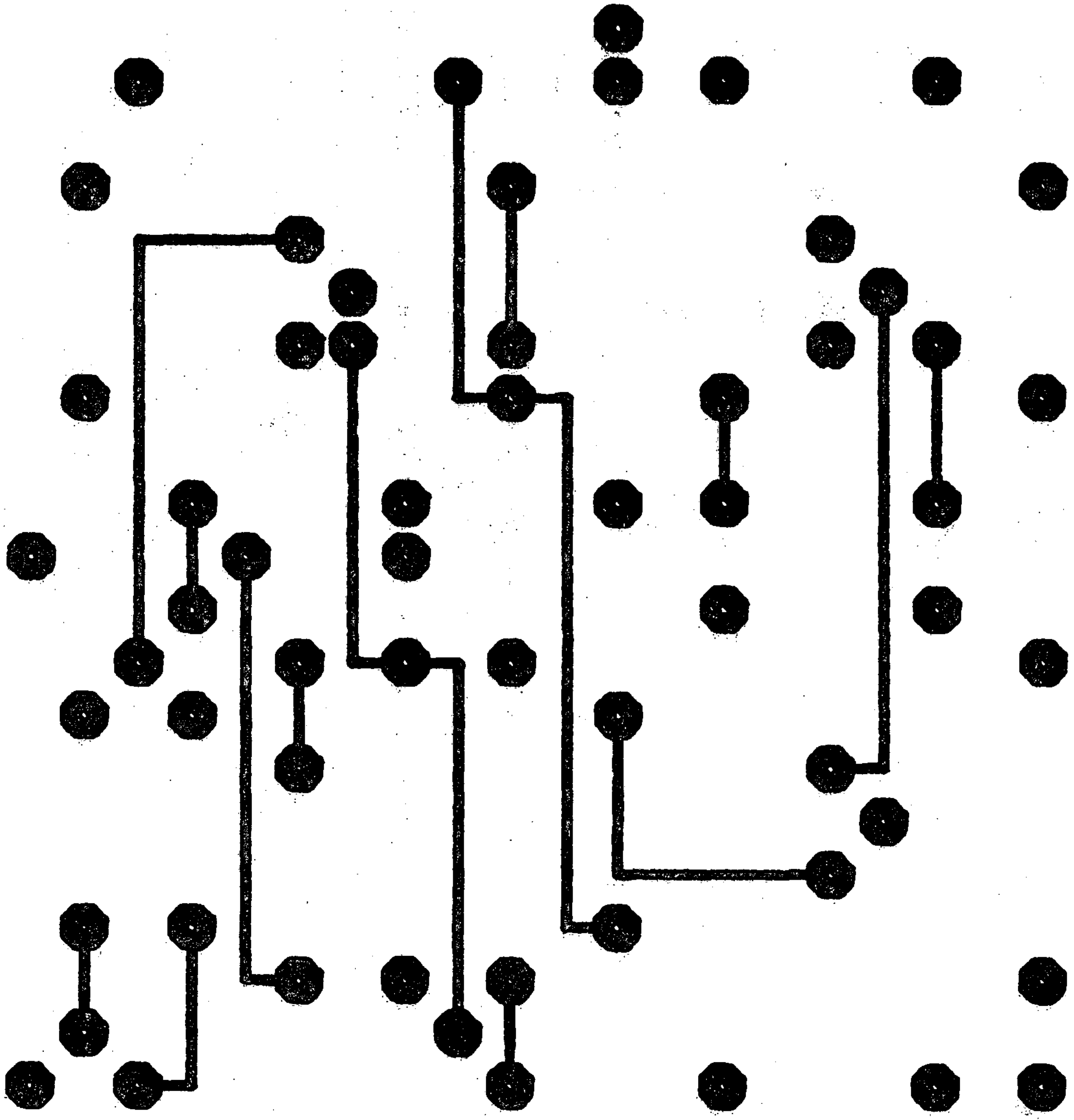
XBL 686-1140

Fig. 15. A graph of the wiring side. Note that most of the wires are horizontal.



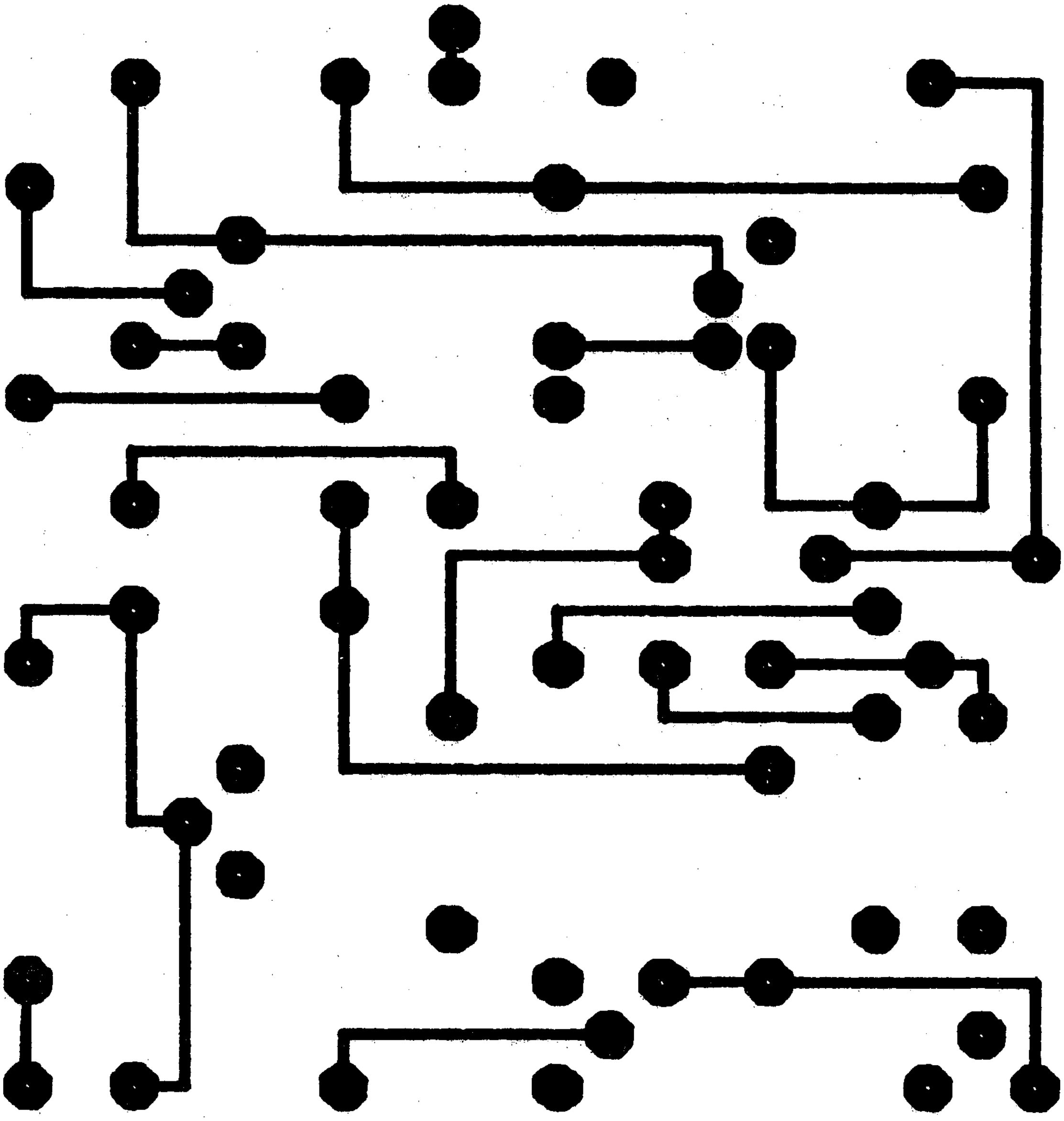
XBL 686-1141

Fig. 16. This shows a composite graph of the same board shown in the previous figures without a board outline.



XBL 686-1142

Fig. 17. The parts side without board outline.



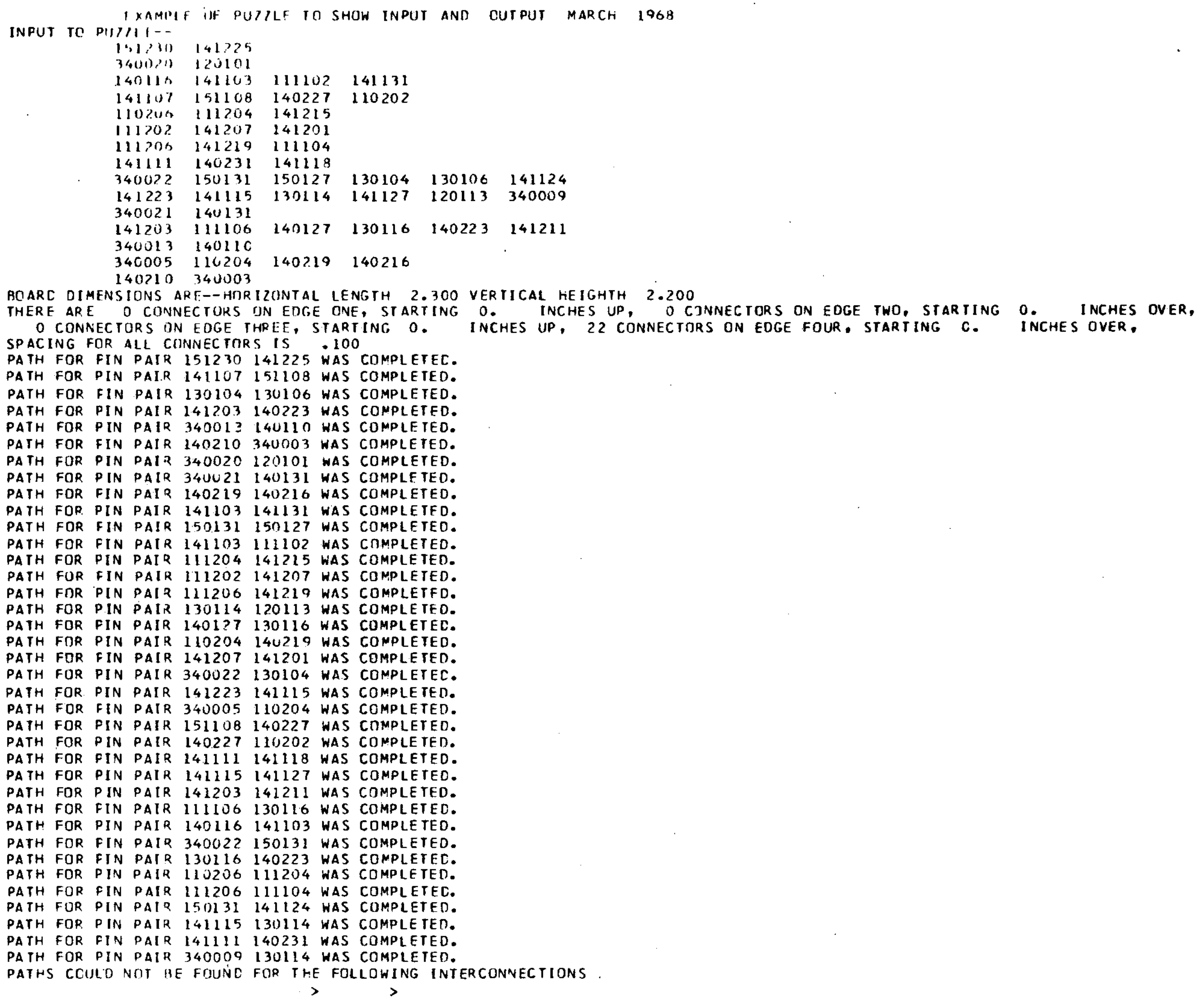
XBL 686-1143

Fig. 18. The wiring side without board outline.

Table I. Data list for successful run on board with board outline.

```
  EXAMPLE OF PUZZLE TO SHOW INPUT AND  OUTPUT  MARCH  1968 2       .1       .1
    2.3     2.2      0     0.0      0     0.0      0     0.0      22    0.0      .1
151230 141225
340020 120101
140116 141103 111102 141131
141107 151108 140227 110202
110206 111204 141215
111202 141207 141201
111206 141219 111104
141111 140231 141118
340022 150131 150127 130104 130106 141124
141223 141115 130114 141127 120113 340009
340021 140131
141203 111106 140127 130116 140223 141211
340013 140110
340005 110204 140219 140216
140210 340003
```

Table II. Computer printout of PUZZLE-interpreted data and summary of action by PUZZLE.



```
                EXAMPLE OF PUZZLE TO SHOW INPUT AND  OUTPUT  MARCH  1968
INPUT TO PUZZLE--
          151230  141225
          340020  120101
          140116  141103  111102  141131
          141107  151108  140227  110202
          110206  111204  141215
          111202  141207  141201
          111206  141219  111104
          141111  140231  141118
          340022  150131  150127  130104  130106  141124
          141223  141115  130114  141127  120113  340009
          340021  140131
          141203  111106  140127  130116  140223  141211
          340013  140110
          340005  110204  140219  140216
          140210  340003
BOARC DIMENSIONS ARE--HORIZONTAL LENGTH  2.300 VERTICAL HEIGHTH  2.200
THERE ARE   0 CONNECTORS ON EDGE ONE, STARTING  0.    INCHES UP,   0 CONNECTORS ON EDGE TWO, STARTING  0.    INCHES OVER,
   0 CONNECTORS ON EDGE THREE, STARTING  0.    INCHES UP,  22 CONNECTORS ON EDGE FOUR, STARTING  C.    INCHES OVER,
SPACING FOR ALL CONNECTORS IS   .100
PATH FOR FIN PAIR 151230 141225 WAS COMPLETEC.
PATH FOR PIN PAIR 141107 151108 WAS COMPLETED.
PATH FOR FIN PAIR 130104 130106 WAS COMPLETED.
PATH FOR PIN PAIR 141203 140223 WAS COMPLETED.
PATH FOR PIN PAIR 340013 140110 WAS COMPLETED.
PATH FOR FIN PAIR 140210 340003 WAS COMPLETED.
PATH FOR PIN PAIR 340020 120101 WAS COMPLETED.
PATH FOR PIN PAIR 340021 140131 WAS COMPLETED.
PATH FOR FIN PAIR 140219 140216 WAS COMPLETED.
PATH FOR PIN PAIR 141103 141131 WAS COMPLETED.
PATH FOR FIN PAIR 150131 150127 WAS COMPLETED.
PATH FOR FIN PAIR 141103 111102 WAS COMPLETED.
PATH FOR PIN PAIR 111204 141215 WAS COMPLETED.
PATH FOR FIN PAIR 111202 141207 WAS COMPLETED.
PATH FOR PIN PAIR 111206 141219 WAS COMPLETED.
PATH FOR PIN PAIR 130114 120113 WAS COMPLETED.
PATH FOR PIN PAIR 140127 130116 WAS COMPLETEC.
PATH FOR PIN PAIR 110204 140219 WAS COMPLETED.
PATH FOR FIN PAIR 141207 141201 WAS COMPLETED.
PATH FOR PIN PAIR 340022 130104 WAS COMPLETEC.
PATH FOR PIN PAIR 141223 141115 WAS COMPLETED.
PATH FOR FIN PAIR 340005 110204 WAS COMPLETED.
PATH FOR PIN PAIR 151108 140227 WAS COMPLETED.
PATH FOR PIN PAIR 140227 110202 WAS COMPLETED.
PATH FOR FIN PAIR 141111 141118 WAS COMPLETED.
PATH FOR PIN PAIR 141115 141127 WAS COMPLETED.
PATH FOR PIN PAIR 141203 141211 WAS COMPLETED.
PATH FOR FIN PAIR 111106 130116 WAS COMPLETED.
PATH FOR PIN PAIR 140116 141103 WAS COMPLETED.
PATH FOR FIN PAIR 340022 150131 WAS COMPLETED.
PATH FOR FIN PAIR 130116 140223 WAS COMPLETEC.
PATH FOR PIN PAIR 110206 111204 WAS COMPLETED.
PATH FOR FIN PAIR 111206 111104 WAS COMPLETEC.
PATH FOR PIN PAIR 150131 141124 WAS COMPLETED.
PATH FOR PIN PAIR 141115 130114 WAS COMPLETED.
PATH FOR FIN PAIR 141111 140231 WAS COMPLETED.
PATH FOR PIN PAIR 340009 130114 WAS COMPLETED.
PATHS COULD NOT BE FOUND FOR THE FOLLOWING INTERCONNECTIONS .
                                   >       >
```

Table III. A listing by PUZZLE of the data which it processed.

```
INFUT PROCESSED BY PUZZLE--
           151230  141225
           150103  120101
           140116  141103  111102  141131
           141107  151108  140227  110202
           110206  111204  141215
           111202  141207  141201
           111206  141219  111104
           141111  140231  141118
           150101  150131  150127  130104  130106  141124
           141223  141115  130114  141127  120113  150204
           140101  140131
           141203  111106  140127  130116  140223  141211
           150110  140110
           150208  110204  140219  140216
           140210  150210
```

Table IV. Listing of successful data deck for board without board outline.

```
   EXAMPLE OF PUZZLE TO SHOW INPUT AND  OUTPUT  MARCH  1968 2
151230 141225
150103 120101
140116 141103 111102 141131
141107 151108 140227 110202
110206 111204 141215
111202 141207 141201
111206 141219 111104
141111 140231 141118
150101 150131 150127 130104 130106 141124
141223 141115 130114 141127 120113 150204
140101 140131
141203 111106 140127 130116 140223 141211
150110 140110
150208 110204 140219 140216
140210 150210
```

Table V. Listing of input and action by PUZZLE.

```
                EXAMPLE OF PUZZLE TO SHOW INPUT AND  OUTPUT  MARCH  1968
INPUT TO PUZZLE--
                151230  141225
                150103  120101
                140116  141103  111102  141131
                141107  151108  140227  110202
                110206  111204  141215
                111202  141207  141201
                111206  141219  111104
                141111  140231  141118
                150101  150131  150127  130104  130106  141124
                141223  141115  130114  141127  120113  150204
                140101  140131
                141203  111106  140127  130116  140223  141211
                150110  140110
                150208  110204  140219  140216
                140210  150210
PATH FOR PIN PAIR 151230 141225 WAS COMPLETED.
PATH FOR PIN PAIR 141107 151108 WAS COMPLETED.
PATH FOR PIN PAIR 130104 130106 WAS COMPLETED.
PATH FOR PIN PAIR 140101 140131 WAS COMPLETED.
PATH FOR PIN PAIR 141203 140223 WAS COMPLETED.
PATH FOR PIN PAIR 150110 140110 WAS COMPLETED.
PATH FOR PIN PAIR 140210 150210 WAS COMPLETED.
PATH FOR PIN PAIR 140219 140216 WAS COMPLETED.
PATH FOR PIN PAIR 150103 120101 WAS COMPLETED.
PATH FOR PIN PAIR 141103 141131 WAS COMPLETED.
PATH FOR PIN PAIR 150131 150127 WAS COMPLETED.
PATH FOR PIN PAIR 141103 111102 WAS COMPLETED.
PATH FOR PIN PAIR 111204 141215 WAS COMPLETED.
PATH FOR PIN PAIR 111202 141207 WAS COMPLETED.
PATH FOR PIN PAIR 111206 141219 WAS COMPLETED.
PATH FOR PIN PAIR 130114 120113 WAS COMPLETED.
PATH FOR PIN PAIR 140127 130116 WAS COMPLETED.
PATH FOR PIN PAIR 110204 140219 WAS COMPLETED.
PATH FOR PIN PAIR 141207 141201 WAS COMPLETED.
PATH FOR PIN PAIR 141223 141115 WAS COMPLETED.
PATH FOR PIN PAIR 150208 110204 WAS COMPLETED.
PATH FOR PIN PAIR 151108 140227 WAS COMPLETED.
PATH FOR PIN PAIR 140227 110202 WAS COMPLETED.
PATH FOR PIN PAIR 141111 141118 WAS COMPLETED.
PATH FOR PIN PAIR 150101 130104 WAS COMPLETED.
PATH FOR PIN PAIR 141115 141127 WAS COMPLETED.
PATH FOR PIN PAIR 141203 141211 WAS COMPLETED.
PATH FOR PIN PAIR 111106 130116 WAS COMPLETED.
PATH FOR PIN PAIR 140116 141103 WAS COMPLETED.
PATH FOR PIN PAIR 150127 130104 WAS COMPLETED.
PATH FOR PIN PAIR 130116 140223 WAS COMPLETED.
PATH FOR PIN PAIR 110206 111204 WAS COMPLETED.
PATH FOR PIN PAIR 111206 111104 WAS COMPLETED.
PATH FOR PIN PAIR 150131 141124 WAS COMPLETED.
PATH FOR PIN PAIR 141115 130114 WAS COMPLETED.
PATH FOR PIN PAIR 141111 140231 WAS COMPLETED.
PATH FOR PIN PAIR 130114 150204 WAS COMPLETED.
PATHS COULD NOT BE FOUND FOR THE FOLLOWING INTERCONNECTIONS
                                  >       >
```

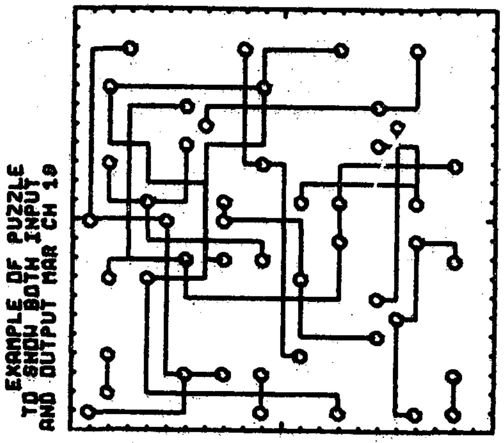
Table VI. Listing of processed input.

```
INPUT PROCESSED BY PUZZLE--
          151230  141225
          340020  120101
          140116  141103  111102  141131
          141107  151108  140227  110202
          110206  111204  141215
          111202  141207  141201
          111206  141219  111104
          141111  140231  141118
          340022  150131  150127  130104  130106  141124
          141223  141115  130114  141127  120113  340009
          340021  140131
          141203  111106  140127  130116  140223  141211
          340013  140110
          340005  110204  140219  140216
          140210  340003
```

APPENDIX 4

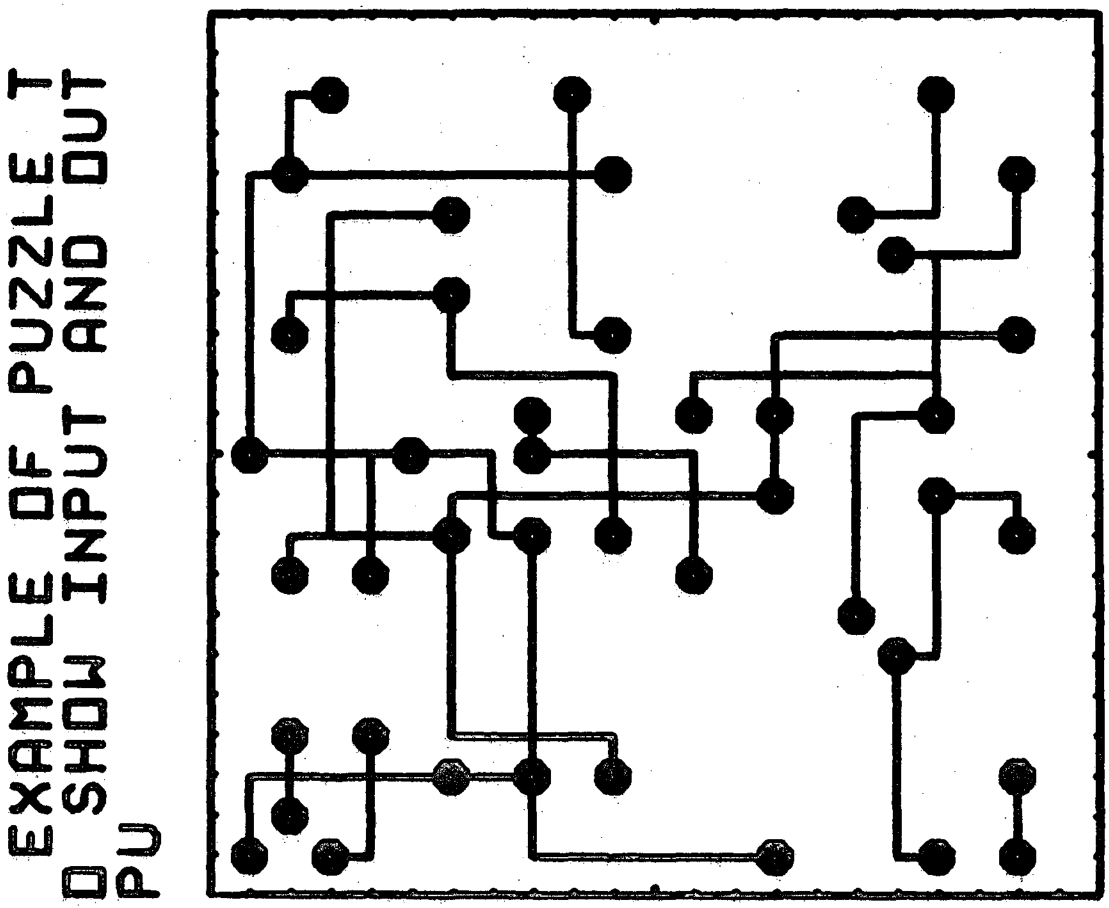
Figures 19 and 20 show the graphic results obtained when various input errors occur. Figure 19 shows a size 1 drawing (actual size) which was rendered thus because there was a faulty size specification on the relevant data card (1968 overlaps this field on the data card listed in Table VII, and Table VIII shows the error message produced by the program). This figure is supposed to be of the same board as shown in Appendix 3, but close examination shows that some pads are missing. There are two causes of missing pads and connections. First, there were excess data on the label card (.3 over the right, Table VII). Hence the program expected the second card to be board outline parameters, and did not list the first pin pair among the data, Table VIII. Secondly, two data points were faulty. Table VIII lists these. The first error is a simple case of trying to use a connector code though none exists. The second error was probably a key-punch error, as 62 is not one of the allowable "square codes." Table IX shows the list of data actually accepted by PUZZLE. Under normal conditions this set of data would produce only one graph, the composite. However, the user employed the force option to get all his graphs. (The additional graphs do not appear in this report.) Table X shows the final list of data processed by PUZZLE. Note that the final card listed in "Input to PUZZLE" is not in the processed list; 340020 was not accepted. On the other hand note that although 146215 was not accepted (third card, third item), the additional connection specified on that card was accepted.

Figure 20 indicates the result of very faulty data. Notice the partial label and missing connections. Table XI shows the data. Some of the bad data are immediately apparent--for example, the card with only the 4 punch. Table XII shows data not accepted by the computer. From a glance at Table XIII one sees that occasionally bad data cannot be conveniently deleted without causing other erroneous conditions. As these are encountered attempts are made to modify the program, but it is impossible to catch all types of errors.



XBL 686-1136

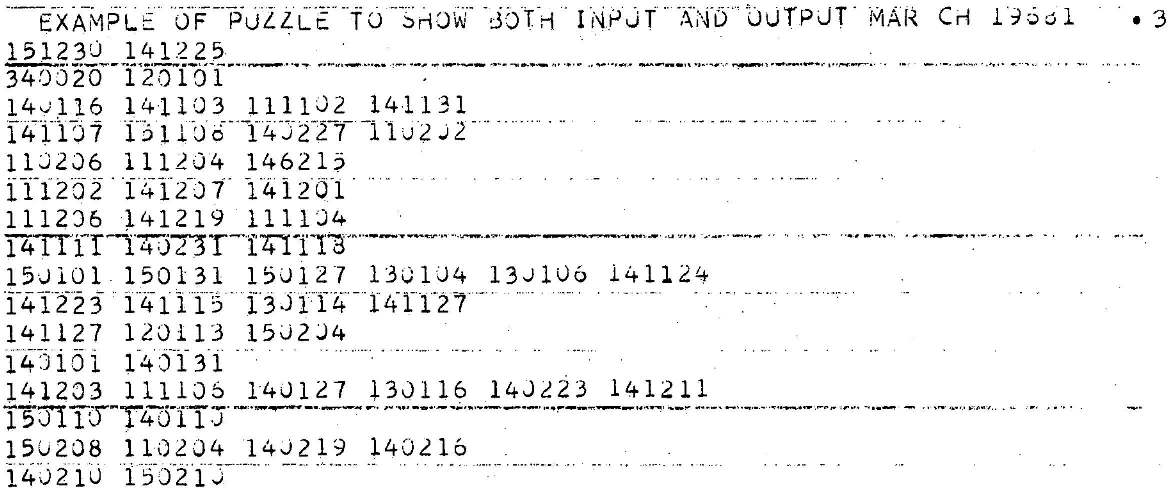
Fig. 19. A composite graph in size one. This is the same board as Figs. 15 through 18, but note that three of the pads referenced by bad data do not appear.



XBL 686-1137

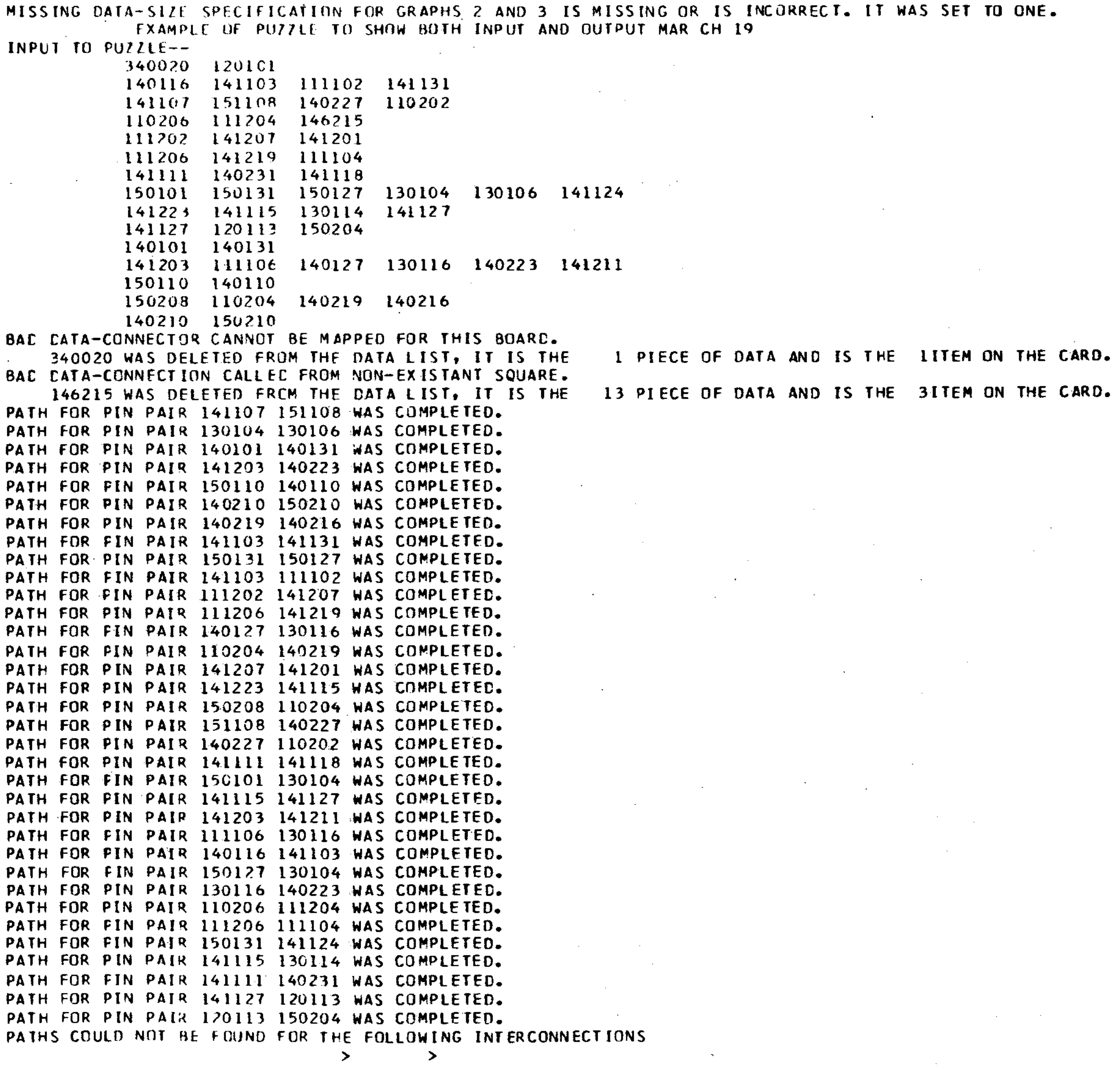
Fig. 20. This is a composite graph of a partial board. The data construction was quite faulty. Notice that the routing has changed significantly from the complete board.

Table VII. Listing of data in improper form. The .3 on the first card was considered a board parameter, and hence the first card (interconnection) was read as board outlines and consequently was ignored because not in proper form.



```
  EXAMPLE OF PUZZLE TO SHOW BOTH INPUT AND OUTPUT MAR CH 19681    .3
151230 141225
340020 120101
140116 141103 111102 141131
141107 151108 140227 110202
110206 111204 146215
111202 141207 141201
111206 141219 111104
141111 140231 141118
150101 150131 150127 130104 130106 141124
141223 141115 130114 141127
141127 120113 150204
140101 140131
141203 111106 140127 130116 140223 141211
150110 140110
150208 110204 140219 140216
140210 150210
```

Table VIII. A listing to show PUZZLE's interpretation of data listed in Table VII.

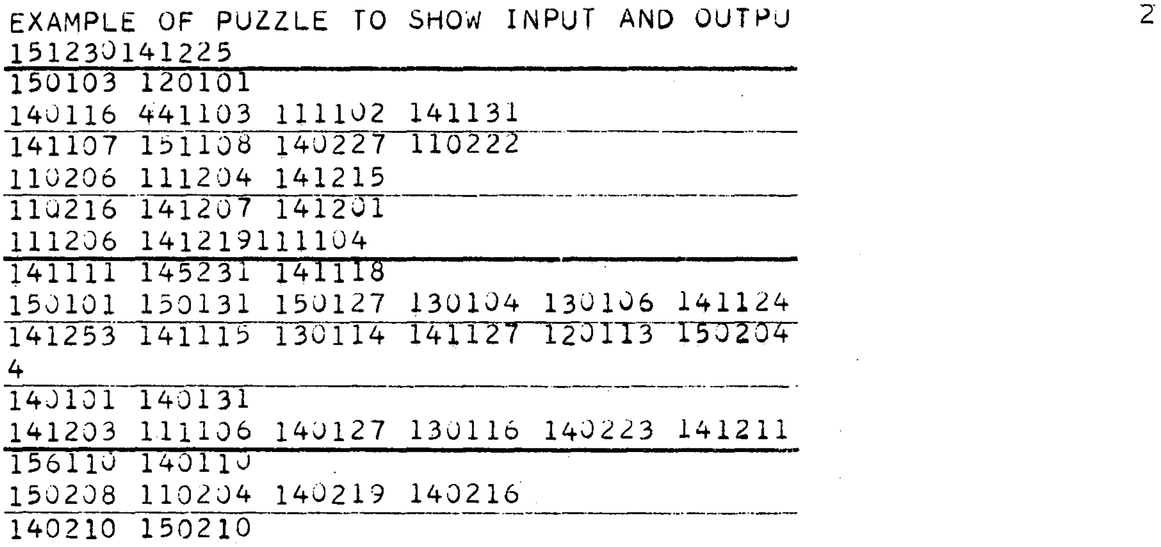


```
MISSING DATA-SIZE SPECIFICATION FOR GRAPHS 2 AND 3 IS MISSING OR IS INCORRECT. IT WAS SET TO ONE.
                 FXAMPLE UF PUZZLE TO SHOW BOTH INPUT AND OUTPUT MAR CH 19
INPUT TO PUZZLE--
                340020  1201C1
                140116  141103  111102  141131
                141107  151108  140227  110202
                110206  111204  146215
                111202  141207  141201
                111206  141219  111104
                141111  140231  141118
                150101  150131  150127  130104  130106  141124
                141223  141115  130114  141127
                141127  120113  150204
                140101  140131
                141203  111106  140127  130116  140223  141211
                150110  140110
                150208  110204  140219  140216
                140210  150210
BAC CATA-CONNECTOR CANNOT BE MAPPED FOR THIS BOARC.
      340020 WAS DELETED FROM THE DATA LIST, IT IS THE     1 PIECE OF DATA AND IS THE  1ITEM ON THE CARD.
BAC CATA-CONNECTION CALLEC FROM NON-EXISTANT SQUARE.
      146215 WAS DELETED FRCM THE DATA LIST, IT IS THE    13 PIECE OF DATA AND IS THE  3ITEM ON THE CARD.
PATH FOR PIN PAIR 141107 151108 WAS COMPLETED.
PATH FOR PIN PAIR 130104 130106 WAS COMPLETED.
PATH FOR PIN PAIR 140101 140131 WAS COMPLETED.
PATH FOR PIN PAIR 141203 140223 WAS COMPLETED.
PATH FOR PIN PAIR 150110 140110 WAS COMPLETED.
PATH FOR PIN PAIR 140210 150210 WAS COMPLETED.
PATH FOR PIN PAIR 140219 140216 WAS COMPLETED.
PATH FOR FIN PAIR 141103 141131 WAS COMPLETED.
PATH FOR PIN PAIR 150131 150127 WAS COMPLETED.
PATH FOR FIN PAIR 141103 111102 WAS COMPLETED.
PATH FOR PIN PAIR 111202 141207 WAS COMPLETED.
PATH FOR PIN PAIR 111206 141219 WAS COMPLETED.
PATH FOR FIN PAIR 140127 130116 WAS COMPLETED.
PATH FOR PIN PAIR 110204 140219 WAS COMPLETED.
PATH FOR PIN PAIR 141207 141201 WAS COMPLETED.
PATH FOR PIN PAIR 141223 141115 WAS COMPLETED.
PATH FOR PIN PAIR 150208 110204 WAS COMPLETED.
PATH FOR PIN PAIR 151108 140227 WAS COMPLETED.
PATH FOR PIN PAIR 140227 110202 WAS COMPLETED.
PATH FOR PIN PAIR 141111 141118 WAS COMPLETED.
PATH FOR FIN PAIR 15C101 130104 WAS COMPLETED.
PATH FOR PIN PAIR 141115 141127 WAS COMPLETED.
PATH FOR PIN PAIR 141203 141211 WAS COMPLETED.
PATH FOR FIN PAIR 111106 130116 WAS COMPLETED.
PATH FOR PIN PAIR 140116 141103 WAS COMPLETED.
PATH FOR FIN PAIR 150127 130104 WAS COMPLETED.
PATH FOR PIN PAIR 130116 140223 WAS COMPLETED.
PATH FOR PIN PAIR 110206 111204 WAS COMPLETED.
PATH FOR FIN PAIR 111206 111104 WAS COMPLETED.
PATH FOR FIN PAIR 150131 141124 WAS COMPLETED.
PATH FOR PIN PAIR 141115 130114 WAS COMPLETED.
PATH FOR FIN PAIR 141111 140231 WAS COMPLETED.
PATH FOR PIN PAIR 141127 120113 WAS COMPLETED.
PATH FOR PIN PAIR 120113 150204 WAS COMPLETED.
PATHS COULD NOT BE FOUND FOR THE FOLLOWING INTERCONNECTIONS
                                      >       >
```

Table IX. Listing of data accepted and consequently processed by PUZZLE.

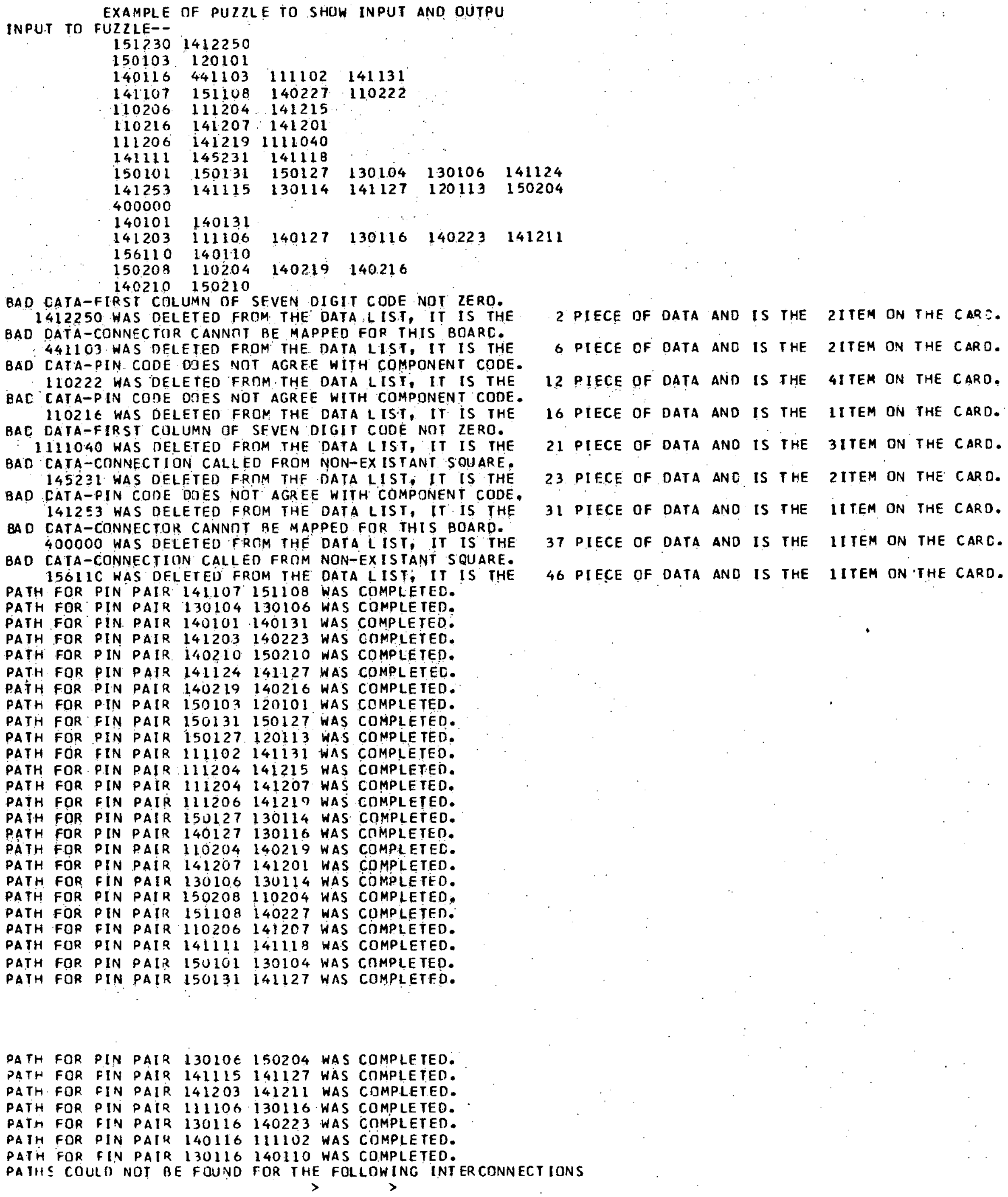
```
INPUT PROCESSED BY PUZZLE--
                 140116  141103  111102  141131
                 141107  151108  140227  110202
                 110206  111204
                 111202  141207  141201
                 111206  141219  111104
                 141111  140231  141118
                 150101  150131  150127  130104  130106  141124
                 141223  141115  130114  141127
                 141127  120113  150204
                 140101  140131
                 141203  111106  140127  130116  140223  141211
                 150110  140110
                 150208  110204  140219  140216
                 140210  150210
```

Table X. Listing of very faulty data. Note card with lone 4.



```
EXAMPLE OF PUZZLE TO SHOW INPUT AND OUTPU                        2
151230141225
150103 120101
140116 441103 111102 141131
141107 151108 140227 110222
110206 111204 141215
110216 141207 141201
111206 141219111104
141111 145231 141118
150101 150131 150127 130104 130106 141124
141253 141115 130114 141127 120113 150204
4
140101 140131
141203 111106 140127 130116 140223 141211
156110 140110
150208 110204 140219 140216
140210 150210
```

Table XI. A listing of interpreted data as well as summary of action on data listed in Table X.



```
         EXAMPLE OF PUZZLE TO SHOW INPUT AND OUTPU
INPUT TO FUZZLE--
           151230 1412250
           150103  120101
           140116  441103  111102  141131
           141107  151108  140227  110222
           110206  111204  141215
           110216  141207  141201
           111206  141219 1111040
           141111  145231  141118
           150101  150131  150127  130104  130106  141124
           141253  141115  130114  141127  120113  150204
           400000
           140101  140131
           141203  111106  140127  130116  140223  141211
           156110  140110
           150208  110204  140219  140216
           140210  150210
BAD CATA-FIRST COLUMN OF SEVEN DIGIT CODE NOT ZERO.
   1412250 WAS DELETED FROM THE DATA LIST, IT IS THE    2 PIECE OF DATA AND IS THE  2ITEM ON THE CARC.
BAD DATA-CONNECTOR CANNOT BE MAPPED FOR THIS BOARC.
    441103 WAS DELETED FROM THE DATA LIST, IT IS THE    6 PIECE OF DATA AND IS THE  2ITEM ON THE CARD.
BAD CATA-PIN CODE DOES NOT AGREE WITH COMPONENT CODE.
    110222 WAS DELETED FROM THE DATA LIST, IT IS THE   12 PIECE OF DATA AND IS THE  4ITEM ON THE CARD.
BAC CATA-PIN CODE DOES NOT AGREE WITH COMPONENT CODE.
    110216 WAS DELETED FROM THE DATA LIST, IT IS THE   16 PIECE OF DATA AND IS THE  1ITEM ON THE CARD.
BAC DATA-FIRST COLUMN OF SEVEN DIGIT CODE NOT ZERO.
   1111040 WAS DELETED FROM THE DATA LIST, IT IS THE   21 PIECE OF DATA AND IS THE  3ITEM ON THE CARD.
BAD CATA-CONNECTION CALLED FROM NON-EXISTANT SQUARE.
    145231 WAS DELETED FROM THE DATA LIST, IT IS THE   23 PIECE OF DATA AND IS THE  2ITEM ON THE CARD.
BAD CATA-PIN CODE DOES NOT AGREE WITH COMPONENT CODE.
    141253 WAS DELETED FROM THE DATA LIST, IT IS THE   31 PIECE OF DATA AND IS THE  1ITEM ON THE CARD.
BAD CATA-CONNECTOR CANNOT BE MAPPED FOR THIS BOARD.
    400000 WAS DELETED FROM THE DATA LIST, IT IS THE   37 PIECE OF DATA AND IS THE  1ITEM ON THE CARC.
BAD CATA-CONNECTION CALLED FROM NON-EXISTANT SQUARE.
    15611C WAS DELETED FROM THE DATA LIST, IT IS THE   46 PIECE OF DATA AND IS THE  1ITEM ON THE CARD.
PATH FOR PIN PAIR 141107 151108 WAS COMPLETED.
PATH FOR PIN PAIR 130104 130106 WAS COMPLETED.
PATH FOR PIN PAIR 140101 140131 WAS COMPLETED.
PATH FOR PIN PAIR 141203 140223 WAS COMPLETED.
PATH FOR PIN PAIR 140210 150210 WAS COMPLETED.
PATH FOR PIN PAIR 141124 141127 WAS COMPLETED.
PATH FOR PIN PAIR 140219 140216 WAS COMPLETED.
PATH FOR PIN PAIR 150103 120101 WAS COMPLETED.
PATH FOR FIN PAIR 150131 150127 WAS COMPLETED.
PATH FOR PIN PAIR 150127 120113 WAS COMPLETED.
PATH FOR FIN PAIR 111102 141131 WAS COMPLETED.
PATH FOR PIN PAIR 111204 141215 WAS COMPLETED.
PATH FOR PIN PAIR 111204 141207 WAS COMPLETED.
PATH FOR FIN PAIR 111206 141219 WAS COMPLETED.
PATH FOR PIN PAIR 150127 130114 WAS COMPLETED.
PATH FOR PIN PAIR 140127 130116 WAS COMPLETED.
PATH FOR PIN PAIR 110204 140219 WAS COMPLETED.
PATH FOR PIN PAIR 141207 141201 WAS COMPLETED.
PATH FOR FIN PAIR 130106 130114 WAS COMPLETED.
PATH FOR PIN PAIR 150208 110204 WAS COMPLETED.
PATH FOR PIN PAIR 151108 140227 WAS COMPLETED.
PATH FOR FIN PAIR 110206 141207 WAS COMPLETED.
PATH FOR PIN PAIR 141111 141118 WAS COMPLETED.
PATH FOR PIN PAIR 150101 130104 WAS COMPLETED.
PATH FOR PIN PAIR 150131 141127 WAS COMPLETED.




PATH FOR PIN PAIR 130106 150204 WAS COMPLETED.
PATH FOR FIN PAIR 141115 141127 WAS COMPLETED.
PATH FOR FIN PAIR 141203 141211 WAS COMPLETED.
PATH FOR PIN PAIR 111106 130116 WAS COMPLETED.
PATH FOR FIN PAIR 130116 140223 WAS COMPLETED.
PATH FOR PIN PAIR 140116 111102 WAS COMPLETED.
PATH FOR FIN PAIR 130116 140110 WAS COMPLETED.
PATHS COULD NOT BE FOUND FOR THE FOLLOWING INTERCONNECTIONS
                                  >       >
```

Table XII. Data actually processed by PUZZLE.

```
INFUT PROCESSED BY PUZZLE--
          150103  120101
          140116  111102  141131
          141107  151108  140227
          110206  111204  141215  141207  141201
          111206  141219
          141111  141118
          150101  150131  150127  130104  130106  141124  141115  130114  141127  120113  150204
          140101  140131
          141203  111106  140127  130116  140223  141211  140110
          150208  110204  140219  140216
          140210  150210
ALL READABLE DATA HAS BEEN PROCESSED.
```

References

1. James M. Spitze, "Printed Circuit Card Design Automation System," Friden (unpublished), November 14, 1967.
2. L. E. West and D. L. Caskey, "Topographic simulation as an aid to printed circuit board design," Sandia Laboratory, Tech. Rept. SC-RR-66-424, August 1966.
3. C. J. Fisk and D. L. Isett, "'ACCEL' Automated Circuit Card Etching Layout," Sandia Laboratory, Technical Memorandum, SC-TM-65-544, October 1965.
4. M. J. Goldberg and R. H. Schroeder et al., Development of On-line System for Computer Aided Design of Integral Circuits, vol. 1, Norden Division of United Aircraft Corp., Air Force Materials Laboratory, Research and Technology Division, Wright-Patterson AFB, Ohio, October 1967.
5. D. A. Wilber and R. Zane, "Development of 'PUZZLE': A program to aid in printed circuit layout design," Lawrence Radiation Laboratory, EE1215, UCID-3079, July 28, 1967.
6. R. Zane, "PUZZLE/Version 4: Operational description," Lawrence Radiation Laboratory, EE1219, UCID-3081, August 9, 1967.
7. Ronald Zane and Deanna A. Wilber, "PUZZLE: A program for computer aided design of printed circuit artwork," Lawrence Radiation Laboratory UCRL-17814, September 12, 1967.

References

1. James M. Spitze, "Printed Circuit Card Design Automation System," Friden (unpublished), November 14, 1967.
2. L. E. West and D. L. Caskey, "Topographic simulation as an aid to printed circuit board design," Sandia Laboratory, Tech. Rept. SC-RR-66-424, August 1966.
3. C. J. Fisk and D. L. Isett, " 'ACCEL' Automated Circuit Card Etching Layout," Sandia Laboratory, Technical Memorandum, SC-TM-65-544, October 1965.
4. M. J. Goldberg and R. H. Schroeder et al., Development of On-line System for Computer Aided Design of Integral Circuits, vol. 1, Norden Division of United Aircraft Corp., Air Force Materials Laboratory, Research and Technology Division, Wright-Patterson AFB, Ohio, October 1967.
5. D. A. Wilber and R. Zane, "Development of 'PUZZLE': A program to aid in printed circuit layout design," Lawrence Radiation Laboratory, EE1215, UCID-3079, July 28, 1967.
6. R. Zane, "PUZZLE/Version 4: Operational description," Lawrence Radiation Laboratory, EE1219, UCID-3081, August 9, 1967.
7. Ronald Zane and Deanna A. Wilber, "PUZZLE: A program for computer aided design of printed circuit artwork," Lawrence Radiation Laboratory UCRL-17814, September 12, 1967.

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or

B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.