A ROUTING PROGRAM FOR MULTILAYER PRINTED CIRCUIT BOARD

by

C. S. Horng and B. S. Ting

Memorandum No. UCB/ERL M77/35

26 May 1977

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A ROUTING PROGRAM FOR MULTILAYER PRINTED CIRCUIT BOARD

C. S. Horng and B. S. Ting

Department of Electrical Engineering and Computer Sciences
and the Electronics Research Laboratory
University of California, Berkeley, California 94720

ABSTRACT

This paper presents the implementation of a system of programs to handle multilayer printed circuit (MPC) board routing problems. A large number of circuit components are mounted on top of the board. Conductor wires are to be laid on various layers on the board such that circuit connections can be properly made. When there are constraints on the size of the board and the width of conductor wires, the routing problem is in general very complicated and difficult. The routing system presented here is based on Ting's thesis [1].

The routing problem is decomposed into several sequential parts and each part is solved separately. Sketches of the implementation are presented along with testing samples to illustrate the use of the system.

I. INTRODUCTION [1,3]

With the advancement of integrated circuit technology, it is now possible to design and fabricate complex circuits within a small chip. Much more complex, large scaled electronic systems can be designed using semiconductor chips as basic building blocks. Multilayer printed circuit (MPC) board is in general the vehicle and the medium for carrying and connecting those building blocks. The component blocks are mounted on top of the board where the terminals are inserted through the drilled-through holes, called pins. Connections among blocks are made by way of printed wires. In order to connect wires on different layers, plated-through holes, called vias, are provided for interlayer connections.

There are two objectives in the routing problem. One is to achieve routability. In other words, we want to make sure that all connections can be properly made. Printed wires are to be laid on each layer such that the circuit connection specifications are implemented by physical conductor wires. The other objective is the need to achieve high circuits density on the board. That is, a large number of circuit components must be mounted on the board. Correspondingly this means more congested space for the printed conductor wires.

This paper deals with a routing system proposed in [1]. The basic philosophy of the approach is to decompose a set of connection specifications in such a way that each connection is consisted of a set of horizontal and vertical conductor segments. No diagonal connections are allowed.

We shall illustrate the routing problem with the specific example in Fig. 1.1. Fig. 1.1 shows a board with 5 signal nets where pins are situated at various locations on the board. A signal net, or simply a net, is a set of equipotential points to be connected together by a continuous wire. The symbol "x" represents a pin and "o", a via. We may assume for now that the board has two layers available for routing. Fig. 1.1b shows one possible realization for the 5 nets in Fig. 1.1a. A realization is the physical routes specifications such that all the nets are connected and no two nets intersect each other. We can observe that three steps are involved in obtaining Fig. 1.1b from Fig. 1.1a. First the pins and vias connection specifications for each net are determined. For instance, net 3 consisting of pins at coordinates (A,1), (A,2), (B,7) are connected through vias at coordinates (A,4), (B,4) to form the connection pattern (A,1)-(A,2)-(A,4)-(B,4)-(B-7). The second step is the assignment of connection patterns to different layers. The solid lines in Fig. 1.1b indicate connections in the first layer and the dotted lines, second layer. The third step we can extract from examining Fig. 1.1b is that the physical realization of the connection patterns must stay within both sides of a line (a line may be a row or a column on the board). Furthermore, two different nets must not intersect each other.

There are 3 implicit assumptions in the realization of Fig. 1.1b. Firstly, the geometries of the pins and vias are at fixed locations. With standard sized IC chips mounted on the board, this first assumption does not impose much restriction on the problem structure. The other two implicit assumptions are due to So [4] and we adopt them for strategical purpose:

(E-1) Only pins and/or vias on the same line (row or column) can be connected directly and the physical routes must be confined within the channels on both sides of the line.

(E-2) All horizontal connections are in one layer and all vertical connections in another layer.

This is called the unidirectional-layer routing by So.

Remark 1.1:

It should be noted here that the two constraints are solely based on strategical considerations because they rule out the necessity of considering various possible alternatives in routing. Furthermore, the schemes are basically economic and can handle many simple circuits routing with two layers only. Finally, as we have seen, the general multilayer routing problem has essentially been reduced to several single line routing problems.

Eleven horizontal tracks and nine vertical tracks are present in the realization in Fig. 1.1b. This track requirement is important because it determines the routing area requirement on the board. If we can specify routability in terms of track requirement, then we can achieve both routability and high circuits density objectives by minimizing track requirement on the board.

Nets 1 and 2 use the vias in first vias column of Fig. 1.1b. However, there is no particular reason why this first vias column should be at the location shown. We may interchange the geometric locations of, say, the first and third vias columns. This is shown in Fig. 1.1c. A reduction of 3 horizontal tracks is achieved by this simple exchange. Thus we have discovered a new step in the decomposition of the routing problem, namely, the geometric placement of the interconnection patterns by permuting vias columns or even pins columns. We have previously observed three steps of decomposition leading to the realization of connections for the board in Fig. 1.1. Vias are first assigned to form connections with the pins. The connection patterns are then assigned to different layers to assist routing the layering step. All the connection patterns are then realized on a single line basis. The geometric placement step follows the vias assignment phase and precedes the layering phase.

We have systematically separated the routing problem into four parts: vias assignment, geometric placement, layering, and single line connection patterns realization. The first two parts may be applied repeatedly in the sense that instead of applying them to the whole board, we may apply them to different sections on the board. We can then merge all the sections together and apply the two steps to complete the connections. A better routing space utilization can be achieved by this scheme.

In Chapter 2 we give a description of the current implementation of the routing system. Chapter 3 contains some results on two test cases, followed by a conclusion in Chapter 4.

1. THE IMPLEMENTATION OF THE ROUTING SYSTEM

According the the approach outlined in the Introduction (more detailed exposition can be found in [1,3]), the routing system is decomposed into three major modules: vias assignment, geometric (linear) placement, and single line routing and plotting (layering is not implemented in the current version, but it can be added as an independent module.) Each module is scheduled as a separate computer run, and the results produced by the earlier modules are used by the latter stages. The structure of the process is depicted by the flowchart in Table 2.1. The modules are briefly described in Section 2.1.

In addition to the major modules mentioned above, there are some small utility routines. These are used primarily in the sectioned application for data preparation and manipulation purposes. The procedure for sectioned application and the ancillary utility functions are the subject of Section 2.2.

2.1 Major Modules of the Routing System

2.1.1 Vias Assignment

Given a board configuration and nets specifications, the purpose of vais assignment is to determine, for each net, the set of vias needed to connect all the pins in it. There are three activities involved in this process, as shown by the flowchart in Table 2.2:

(A) Converting each net into a set of generalized pins;

(b) Preprocessing: reduction and ordering;

(C) Vias assignment.

Each of these steps is described separately in the following.

(A) Converting nets specifications into sets of generalized pins

Given a board configuration, i.e. numbers of pin rows and columns, and a set of nets specifications, this step converts each net into a set of "generalized pins."

<u>Board Configuration and Nets Specifications</u>

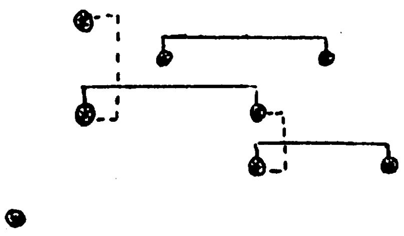
The board configuration is specified by giving the number NX of pin columns and the number NY of pin rows. Rows and columns are numbered as follows:

```
NY  *   *   *   .   .   .   *   *
 .  *   *   *   .   .   .   *   *
 .  .   .   .       .       .   .
 .  .   .   .       .       .   .
 .  .   .   .       .       .   .
 3  *   *   *   .   .   .   *   *
 2  *   *   *   .   .   .   *   *
 1  *   *   *   .   .   .   *   *
    1   2   3   .   .   .   .   NX
```

For each net the net specification consists of the coordinates of the pins contained in that net.

Generalized Pins

For vias assignment purpose, each net is first converted into a set of "generalized pins" (GP's). A generalized pin is simply a set of pins which can be connected (under the type of connection constraints adopted) without using vias. Two pins are considered to be in the same GP if and only if they have either the same X-coordinate or the same Y-coordinate. Also, the relation is transitive, i.e., if pins A and B are in the same GP, and pins B and C are in the same GP, then pins A and C also belong to the same GP. An example of a net and the set of GP's representing it is given below:



In the above example there are 8 pins. These are converted into 3 generalized pins as indicated by the connections.

Determining the set of pins belonging to the same GP

In order to convert the pins in a net into a set of generalized pins, we can define a binary relation G on the set of pins in the net. Two pins A and B in the same net satisfy the relation, written as G(A,B), if and only if A and B have either the same X-coordinate or the same Y-coordinate. It is easily seen that the reflexive, transitive closure G* of the relation G defines the relation: "is in the same GP as". To determine which and how many pins belong to a GP, we first construct the matrix representation of the relation G. Then the relfexive transitive closure G* is computed using the algorithm due to Warshall [5].

(B) Preprocessing

Before the vias assignment is actually performed, two heuristic preprocessing steps are carried out aiming at improving the vias assignment operation. The first involves a "reduction" in which each generalized pin is reduced into a "representative" pin. In the vias expansion only those representatives are considered for connection. The second type of preprocessing is an "ordering" which determines the sequence in which the nets are processed in the expansion operation. Two options are available in determing that sequence.

Reduction

The two preprocessing operations are based on the idea illustrated by Table 2.3. Here the columns are headed by pin row numbers and the rows are headed by net numbers. For each net i, if pin row number j appears in the net and is assigned generalized pin number k, then we put k in entry (i,j). For each column we count the number of non-empty entries, and put the number in the corresponding entry of the vector COUNT.

The purpose of reduction is to reduce each generalized pin in a net into one "representative" pin row, which is chosen as the one with the minimum COUNT number. For example, consider net 1 in Table 2.3. Here generalized pin 1 consists of two pin row numbers, 1 and 4, with count values 5 and 7, respectively. In this case, pin row 1 will be selected as the representative for GP 1, i.e., only this pin row will be considered for vias connection. After this, entry (1,4) will be cleared to 0, and the corresponding COUNT number decremented by 1.

Similarly, GP 2 consists of three pin rows, 3, 6, and 11, with COUNT values 8, 6, 6, respectively. In case of a tie, the breaking policy is arbitrary. If we choose pin row 6, then entries (1,3) and (1,11) will be cleared and the COUNT numbers decremented accordingly.

The process is repeated for all the nets. In our implementation we rearrange all the pin row numbers in each GP to make the chosen representative the first element for that GP. Thus only the first pin row number in each GP is considered for vias connection.

Ordering

After the reduction process we have a reduced COUNT vector. The ordering operation is to determine a sequence in which the nets are considered for vias assignment. Two options are available for this purpose, MAXIMUM or MINIMUM. For the MAXIMUM case we pick the pin row with the largest residue COUNT number, and all those nets having this row number as their representative are expanded first. For the MINIMUM case we choose the opposite.

In our testing examples we use exclusively the MAXIMUM option, since this is considered a better heuristic.

(C) Vias Assignment

With the order in which the nets are to be considered for expansion and the representative pin row for each GP determined, the nets can now be expanded one by one. The procedure follows essentially the algorithm presented in [1], p. 30. In the implementation we add one option allowing a user to specify the maximum number of vias columns to be used in connecting the nets. With such a constraint, it may happen that not all the nets can be completely connected. This feature may be useful under certain circumstances.

The result of the vias assignment is a specification, for each net, of the vias that are included in the tree realizing the connection for that net. Also generated are some summarizing statistics indicating the number of generalized pins, the number of vias columns needed, the number of vias actually used,..., etc.

2.1.2 Geometric Placement

Following the vias assignment stage this part of the process tries to permute the pins and vias columns in order to minimize the number of tracks needed in the final routing. The algorithm implemented is the C-algorithm in [1,2].

To perform the linear placement we have to first convert all the connection patterns into a set of "signals". A signal will be incident with two or more nodes, where a node represents either a vias column, or one or more pin columns which are to be treated as a whole in the linear placement, i.e., they are to be moved together in the permutation.

For example, in Fig. 2.1, we have a board with 8 rows, 8 pin columns, and 2 vias columns. There is a particular net consisting of 8 pins which is realized using 5 vias. Further, assuming that pin columns 1 and 2, 3 and 4, 5 and 6, and 7 and 8 are to be considered as a whole in the movement, then in the formulation for linear placement we will have 6 blocks(nodes). The tree realizing the particular net will contribute 4 to the final list of signals, with the corresponding incident nodes shown below:

<u>Signal</u>	<u>Incident nodes</u>
1	1,3,5
2	1,6
3	2,5,6
4	3,5

Notice that the connection path in row 3 is considered as an "internal" signal and is not counted in the list.

In our implementation we take two real situations into account and allow two types of constraints to be imposed. Type 1 constraints specify that certain blocks are to be assigned to some fixed positions. Type 2 constraints specify that certain blocks are to be placed in a predetermined sequence. We assume that these two types of constraints are disjoint, that is, they contain no blocks in common. An example of such constraints is given below:

2 : This means that there are two type 1 constraints.
1 2 3 4 : Block 1 is to be placed at position 2, Block 3 is to be placed at position 4.

3 : This means that there are three blocks involved in type 2 constraints.

4 6 5 : Blocks 4, 6 and 5 are to be placed in that sequence (although not necessarily continguous, i.e., some other blocks may be inserted in between them.)

2.1.3 Converting Trees into Single Line Problems

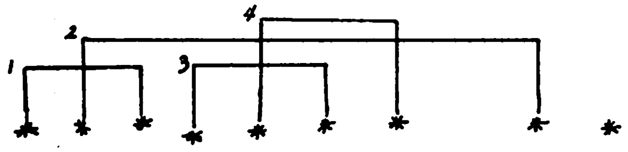
The eventual goal of our approach is to have the original routing problem decompsoed into a set of single line routing problems. After the vias assignment phase we have determined all the trees which properly connect the sets of pins in the nets, using vias wherever necessary. At this point we can generate the set of single line problems. There will be one such problem for each row or column. As an example consider a specific net consisting of 6 pins which are connected using 8 vias, as shown in Fig. 2.2. Notice that the configuration presented is as in the vias assignment phase. This particular tree can be seen to be decomposable into 10 single line nets (6 horizontal ones for rows 2, 3, 4, 5, 7, 10: 4 vertical ones for pin column 4 and vias columns 1, 2, 4.) The only remaining complication is due to the permutation induced by the linear placement. We have to properly locate the position of each pin or via in the final configuration.

The purpose of this part of the process is to take these into consideration and to compute the "contribution" of each net to the single line nets on each row and column. The results are the specifications of the single line nets for all the lines. For each line the nets are specified by giving the net number to which each point on that line belongs.

For example a line with 10 points may be represented by an array of the following 10 numbers:

1, 2, 1, 3, 4, 3, 4, 0, 2, 0

This means that on that line we have the following net configuration:



A "0" means that the particular point is not associated with any net.

2.1.4 Single Line Routing and Plotting

The single line routing and plotting phase is divided into three major steps:

(1) Street assignment: This step determines the order in which the nets are to be routed as well as the specific street assignment for each net;

(2) Routing: Given the sequence and the street assignments, this step actually routes all the nets, i.e., determining the exact track where each connection is to go;

(3) Plotting: Taking the routing results, with appropriate scaling, this step plots the results using a Calcomp plotter.

Options exist for a user to determine to what stage he wants these operations to be carried out. Part of the reason for this arrangement is due to the fact that some of the parameters needed at the later stages can only be determined after the results of the earlier stages are available. The control of this is affected by an "iteration" indicating parameter ITER, and one has the following three options:

(1) ITER = 1 : Perform street assignment only. The purpose of this option is to decide the maximum street congestion among all the single lines. This will enable a user to decide the parameter DIST which determines the horizontal separation between two adjacent node positions, as needed in the routing operation.

(2) ITER = 2 : Street assignment plus routing, but without plotting. Actually performing the routing will enable a user to determine the exact "minimum horizontal separation" (MHS) needed. This is primarily for the purpose of getting a better plotted result.

(3) ITER = 3 : Street assignment, routing, and finally plotting using the Calcomp plotter.

Street Assignments

The street assignment follows essentially the algorithms given in [1]. There are two routines involved in this operation, STASGN and RHOFOR. The former is the ordinary street assignment algorithm and the latter is for the special case when the maximum density $\rho=4$. Given the set of nets on each line, STASGN is first applied to determine the order of routing and street assignment for each net. If $\rho\neq 4$ or if $\rho=4$ but the resulting maximum street congestion M is 2 (Which is the optimal value obtainable.) then the step is completed. Otherwise RHOFOR is applied, in the hope that we can achieve the optimal routing with street congestion equals to 2. If the result of the algorithm indicates that a routing with M=2 is impossible, then the results obtained from STASGN is used.

Routing

With the order in which the nets are to be routed and the street assignment for each net determined, the routing algorithm in [1] (p. 59) is employed to route all the nets. For each net the routing is a two-node-at-a-time, step-by-step forward marching process. The path is extended from one node to the next on the upper or lower street, depending on the specific assignment for that net. When the path is blocked, a switch to the opposite street is made, the path is extended further till the first node position where a return to the originally assigned street is feasible.

In our implementation two local optimizations are included. These have to do with avoiding an immediate, short switch at the beginning of a net or the target of a two-node, forward marching process, as illustrated by Fig. 2.3.

(1) Initial position

If the very first node of a net is immediately blocked by the following node position, then we will eliminate the short segment on the upper street.

(2) Target position

If a switch to the upper street is to be made which will lead immediately to the target, then we again eliminate that short switch.

The examples shown are for the case of routing on the upper street. The case for lower street routing is similar.

Plotting

The final plotting is performed using the graphic system GDS (Graphical Display System [8]). The system is a collection of FORTRAN-callable subroutines which, with the aid of a postprocessor, can generate instructions to control the pen movements of the Calcomp plotter.

Plotting is done line by line. Each line involves essentially two types of activities:

(1) Plotting the node symbols: Two different node symbols are used. Pin positions occupied by chips are indicated by asterisks, no matter whether they are actually used in the connection or not. Vias positions are represented by small circles. However, only those vias actually used in the connection are shown in the plotting.

(2) Plotting all horizontal and vertical line segments.

In addition to the net connections, some annotations can also be plotted. These include the row and column numbers and the other remarks applicable to each particular plot.

2.2 Application with Sectioning

As pointed out in [1] and [3], direct application of the vias assignment algorithm employing vias columns for connection may sometimes result in inefficient or uneven utilization of the available routing space. The approach adopted to improve on this is to divide the original board into several sections and make the connections using "vias rows". However, expansion using vias rows can be achieved by "transposing" the board (or a section of it, if the board is divided) and applying the column expansion algorithm as usual. To follow this approach, we need some data manipulation routines. The process is shown by the flowchart in Table 2.4. The utility functions are described in the following.

2.2.1 Sectioning

For the current implementation sectioning is done by manually deciding the number of sections the board is to be divided into and the section boundaries. Then the sectioning routine is applied which will determine, for each section, the pins falling within the range of that section for each net. As a result some nets in a section may contain no pins at all. After this, each section is treated as a separate problem and the process shown in Table 2.4 is carried out.

2.2.2 Coordinate Transformation

Because of the manner in which the vias assignment problem is formulated, a coordinate transformation is needed if we want to apply the sectioning process. Specifically, the vias expansions are performed by adding vias "columns". On the other hand the major purpose of sectioning is to more fully utilize the space originally unoccupied by physical chips. This in fact requires that we add some vias "rows" to the original configuration. Thus the X- and Y- coordinates have to be interchanged for the pins in each section.

A simplified example may serve to illustrate the points. Suppose we have a 13-row by 6-column configuration as shown in Fig. 2.4a, where chips occupy only those rows indicated by asterisks. Potentially we may add several vias rows at those positions indicted by empty circles before we add vias columns. To fit this into the formulation of our vias assignment algorithm, we need to "compact" the board, i.e., make those occupied rows consecutive, and to "rotate" the board by 90 degrees, i.e., to interchange the new X- and Y-coordinates.

For the example in Fig. 2.4, we have the following correspondences for the original and compacted X- and Y-coordinates:

X:	1	2	3	4	5	6
	1	2	3	4	5	6

Y:	1	2	3	4	5	6	7	8	9	10	11	12	13
	0	1	2	3	4	0	0	0	5	6	7	8	0

In addition we have to interchange the new X- and Y-values. Thus row 1 in the new coordinate system actually corresponds to the "compacted" column 1 in the old system, row 2 to column 2, etc. Point a will have the coordinates (1,2) and (1,1) in the old and new system, respectively.

Later on when we have finished the vias assignment and have decided the positions for those vias columns (in the new coordinate system, which are actually rows in the old system.) to go to, we must again map the coordinates of the added vias onto the original coordinate system. (These vias will be treated as pins when several sections are combined and the whole process is repeated.) At that time another coordinate transformation will be needed.

2.2.3 Recombination

If the original board is divided in the sectioned application, then after all the sections are separately considered for vias assignment and geometric placement, and the coordinates of the pins and vias are properly re-transformed, we have to recombine the sections together. In the expanded nets specifications the vias which are added to each section will be treated essentially like pins.

After the recombination, this expanded version of the nets specifications is then fed through an ordinary, non-sectioned process, as shown in the flowchart in Table 2.4.

3. SAMPLE TESTING AND COMMENTS ON THE IMPLEMENTATION

The system as currently implemented can handle problems of the following size:

Maximum number of rows:	200
Maximum number of pin columns:	60
Maximum number of vias columns:	35
Maximum number of nets:	450

Each net is assumed to have no more than 20 pins and each single line net is assumed to contain no more than 10 nodes.

So far the system has been applied to two sample boards. The first is an ILLIAC IV processing unit board taken from [6]. The second is the board FIRDB given in [7]. The specifications of these two boards and the examples of plotting for the first board are given in the Appendix. Both boards are processed using sectioned and non-sectioned approach. Some summarizing statistics are provided in Table 3.1.

Concerning the implementation itself, there is still too much data handling that has to be done manually. For the system to be more easily applied to problems and to reduce possible human errors, it is necessary to integrate the modules more smoothly such that manual data preparation can be kept to a minimum. Also, depending on the intended usage of the system, soem of the intermediate outputs can probably be eliminated.

4. CONCLUSIONS

We have presented the implementation of a new scheme in solving MPC board routing problems. We have implemented the system to handle two-layer board at the present time, however, it can be readily extended to deal with multilayer cases by including the layering scheme independently.

We have seen a systematic approach in solving the complex routing problem. The system we introduced is quite general in the sense that it can be used to handle some irregularly shaped board routing problem, by properly dividing the board and applying the sectioning scheme iteratively.

5. ACKNOWLEDGEMENTS

The authors are deeply indebted to Professor E. S. Kuh for his continuing encouragement during the course of this research.

Research sponsored by the National Science Foundation Grant ENG74-06651.

REFERENCES

[1] Ting, B.S., A New Approach to the Interconnection Problem of Multilayer Printed Circuit Board, Ph.D. Thesis, Dept. of Elec. Engin. and Comp. Sci., Univ. of Calif., Berkeley, Oct. 1976.

[2] Goto, S., Cederbaum, I., and Ting, B.S., Suboptimal Solution of the Backboard Ordering with Channel Capacity Constraints, Electronics Research Lab. Memo., ERL-M598, Univ. of Calif., Berkeley, July, 1976.

[3] Ting, B.S. and Kuh, E.S., An Approach to the Routing of Multilayer Printed Circuit Boards, Electronics Research Lab. Memo., UCB/ERL M77/7, Univ. of Calif., Berkeley, April 1977.

[4] So, H., "Some Theoretical Results on the Routing of Multilayer, Printed Wiring Boards," Proc. 1974 IEEE International Symp. on Circuits and Systems, pp. 296-303.

[5] Warshall, S., "A Theorem on Boolean Matrices," J. of the ACM, Vol. 9, No. 1 (Jan. 1962), pp. 11-12.

[6] Wu, W. and Schmidt, D.C., "A New Routing Algorithm for Two-sided Boards with Floating Vias," Proc. 13-th Design Automation Conference, June 1976, pp. 151-160.

[7] Stevens, J.E., Jr., Fast Heuristic Techniques for Placing and Wiring Printed Circuit Boards, Ph.D. Thesis, Dept. of Computer Science, Univ. of Illinois, Urbana-Champaign, Oct. 1972 (UIUCDCS-R-72-558).

[8] Graphical Display System, Computer Center, Univ. of Calif., Berkeley, Aug. 1974.

[9] Goto, S. and Kuh, E.S., An Approach to the Two-dimensional Placement Problem in Circuit Layout, Electronics Research Lab. Memo., UCB/ERL M77/16, Univ. of Calif., Berkeley, Mar. 1977.

Appendix

Specifications of the first board

The configuration is as shown in Fig. A.1. There are 20 components on the board, with a total number of 55 signal nets. Edge connectors are transformed into 4 equivalent pin columns. They occupy the fixed position on the right side of the board.

Specifications of the second board

The second board is taken from [7] with slight modification. The specifications are given in Table A.1 and A.2. Table A.1 shows the locations of the components. There are 52 components numbered from A0 to A51 and 15 sets of edge connectors numbered from P1 to P15. This placement is due to Goto and Kuh [9].

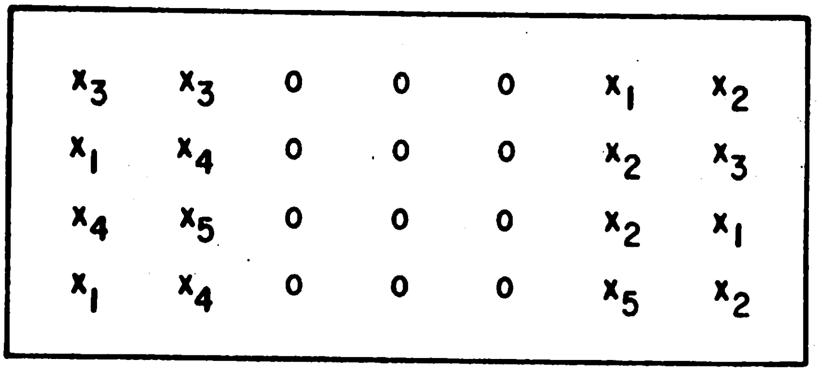
Table A.2 gives the nets specifications. Reading across the line, each row contains four pins. Each pin consists of three fields:

(1) Axxx or Pxxx gives the component to which the pin belongs.

(2) The following 3-digit number gives the location of the pin on that particular component, as shown in Fig. A.7.

(3) The last 1-digit (0 or 1) field indicates the grouping of the pins into signal nets. Each net contains one or more pins with a "0" in this field up to the first one with a "1". For example, net 1 contains 2 pins: P004 003 0 and A000 016 1.

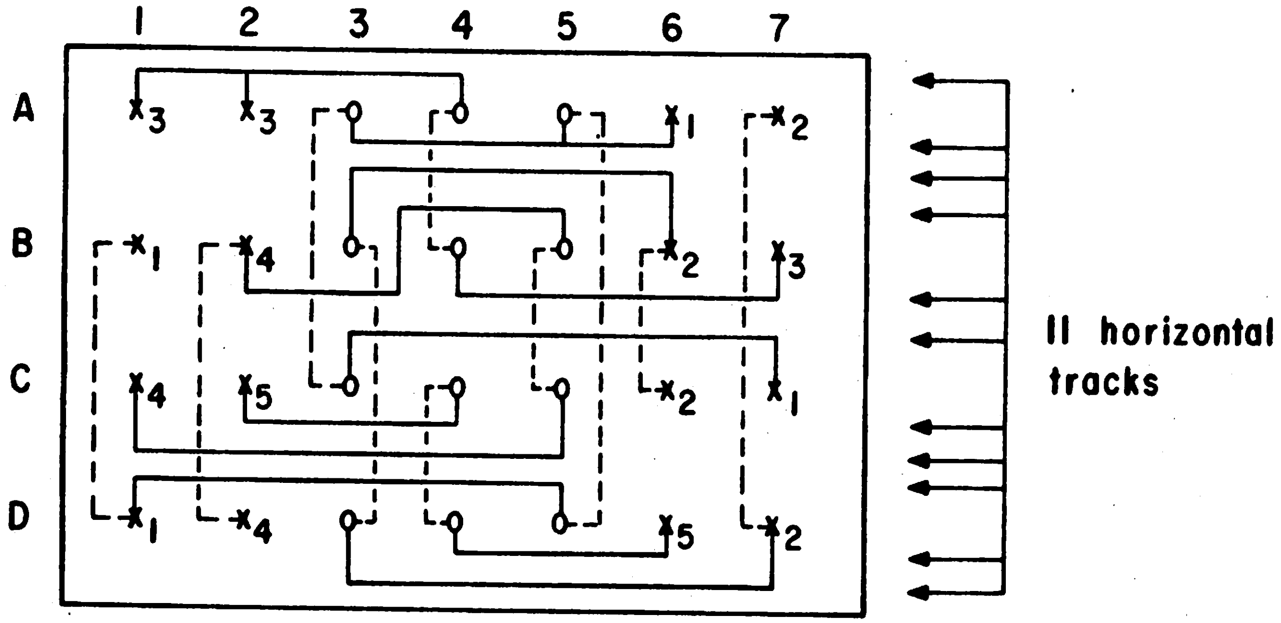
There is one further complication with edge connectors. For the problem specification given in [7] an edge connector may appear twice, i.e., belong to two nets. To resolve the conflicts, we put them into the configuration as in the first board. That is, we use four columns for edge connectors. When a pin first appears, it will be assigned proper position in either column 1 or 3, depending on its exact location. The second occurrence of a pin will be assigned to either column 2 or 4. Thus components representing edge connectors will have the configuration as shown in Fig. A.8.



x_3	x_3	0	0	0	x_1	x_2
x_1	x_4	0	0	0	x_2	x_3
x_4	x_5	0	0	0	x_2	x_1
x_1	x_4	0	0	0	x_5	x_2

(a)

5 nets defined on a two layered board with regular geometry.



(b)

A realization with 11 horizontal tracks and 9 vertical tracks.

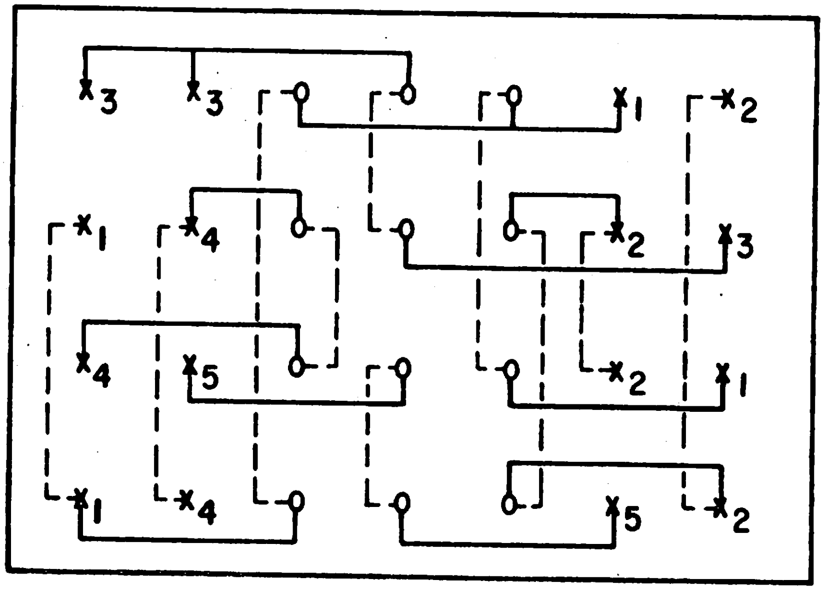


Fig. 1.1

(c)

An alternative realization with 8 horizontal tracks.

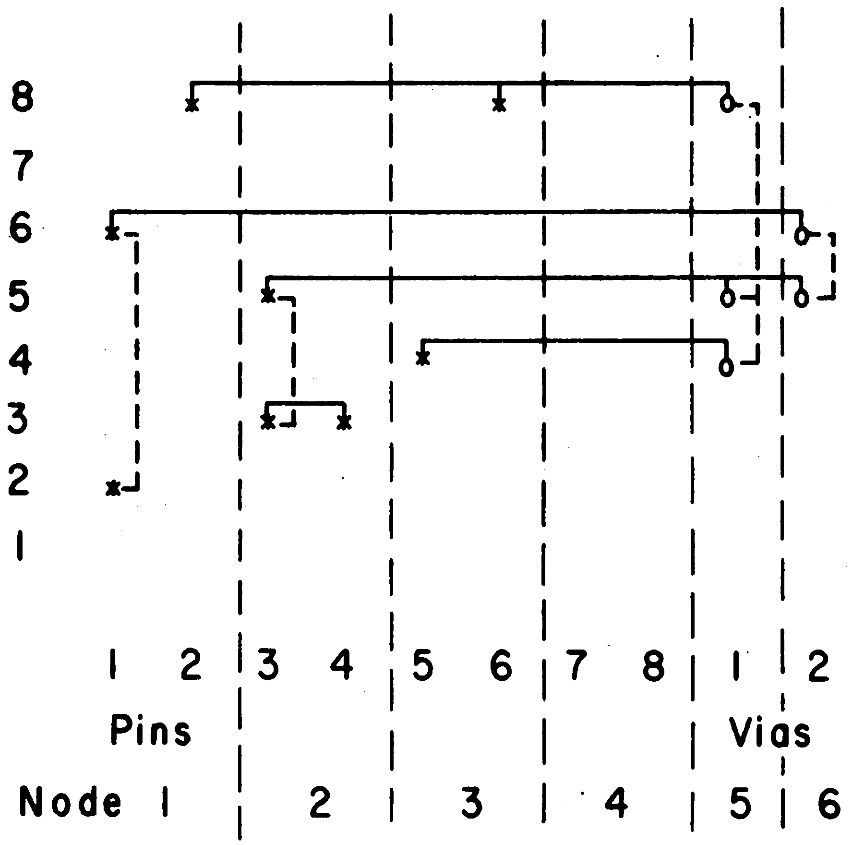


Fig. 2.1

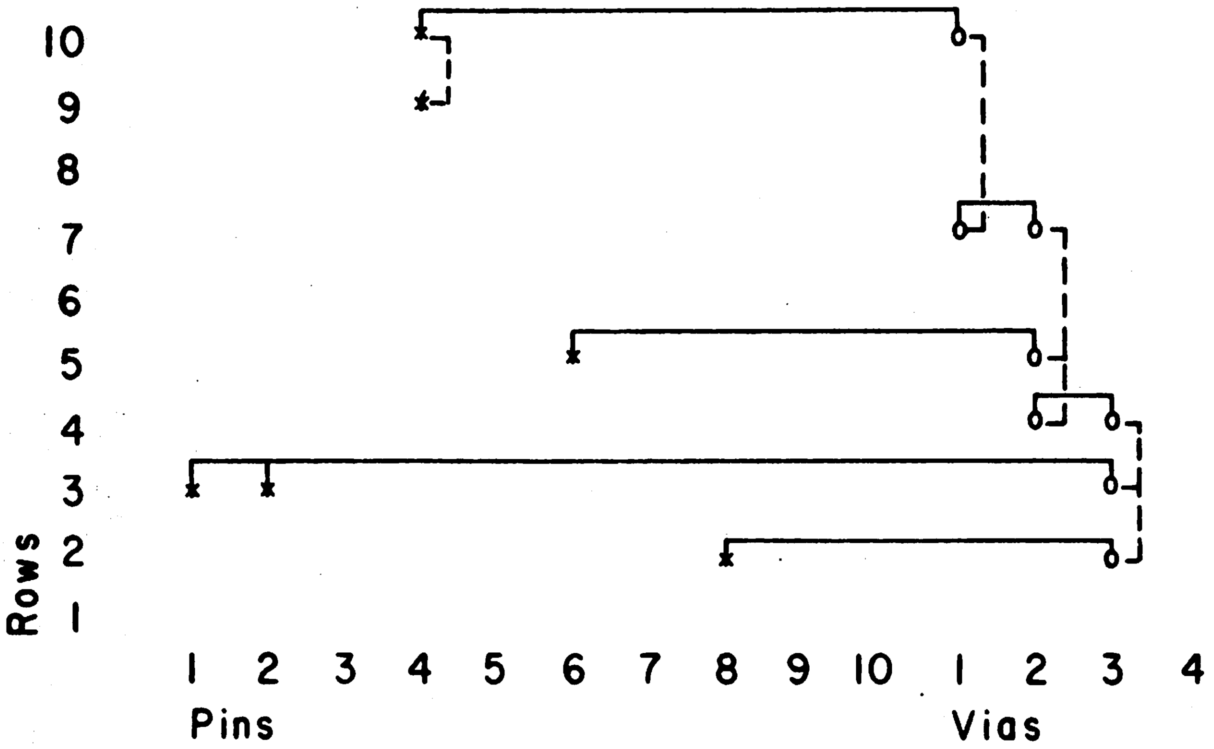
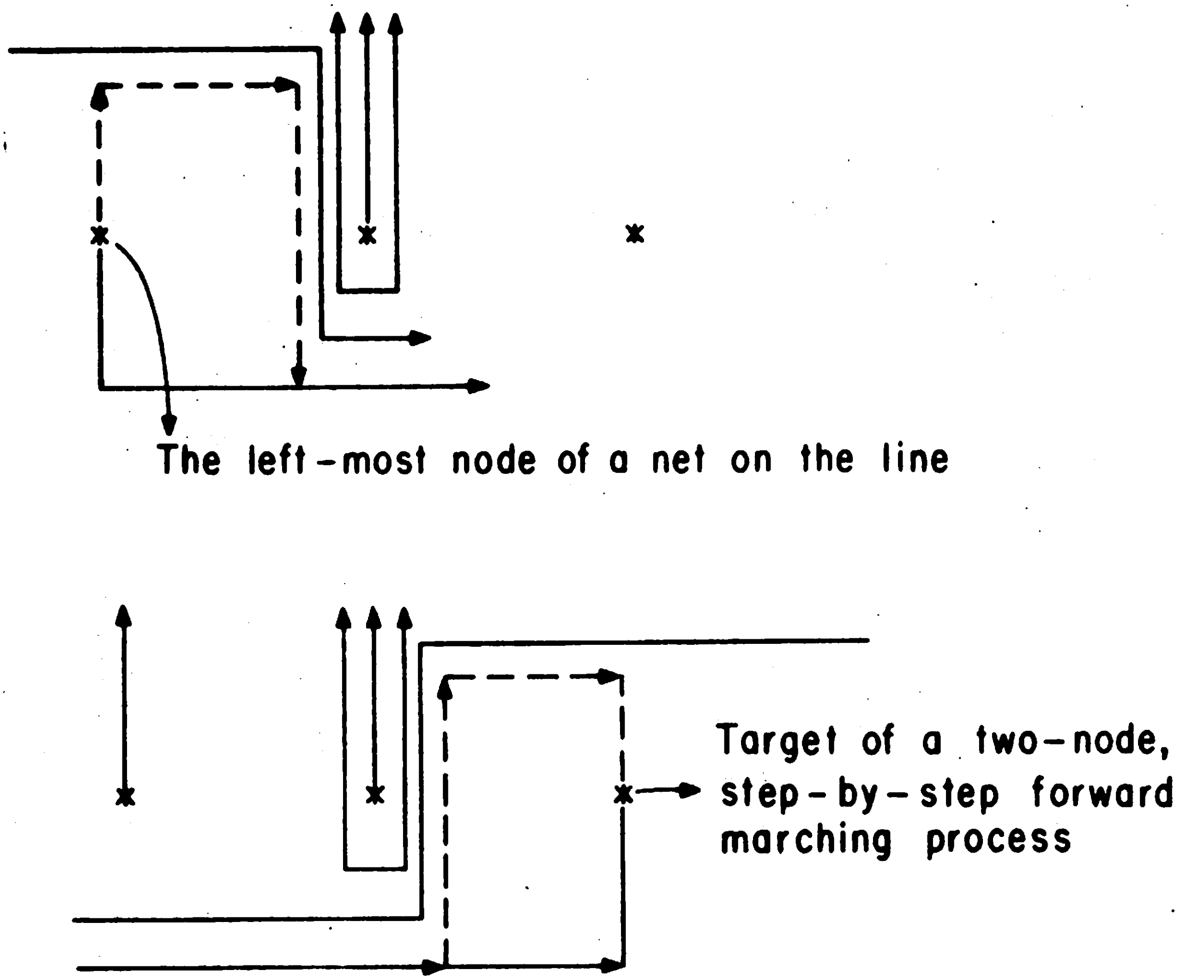


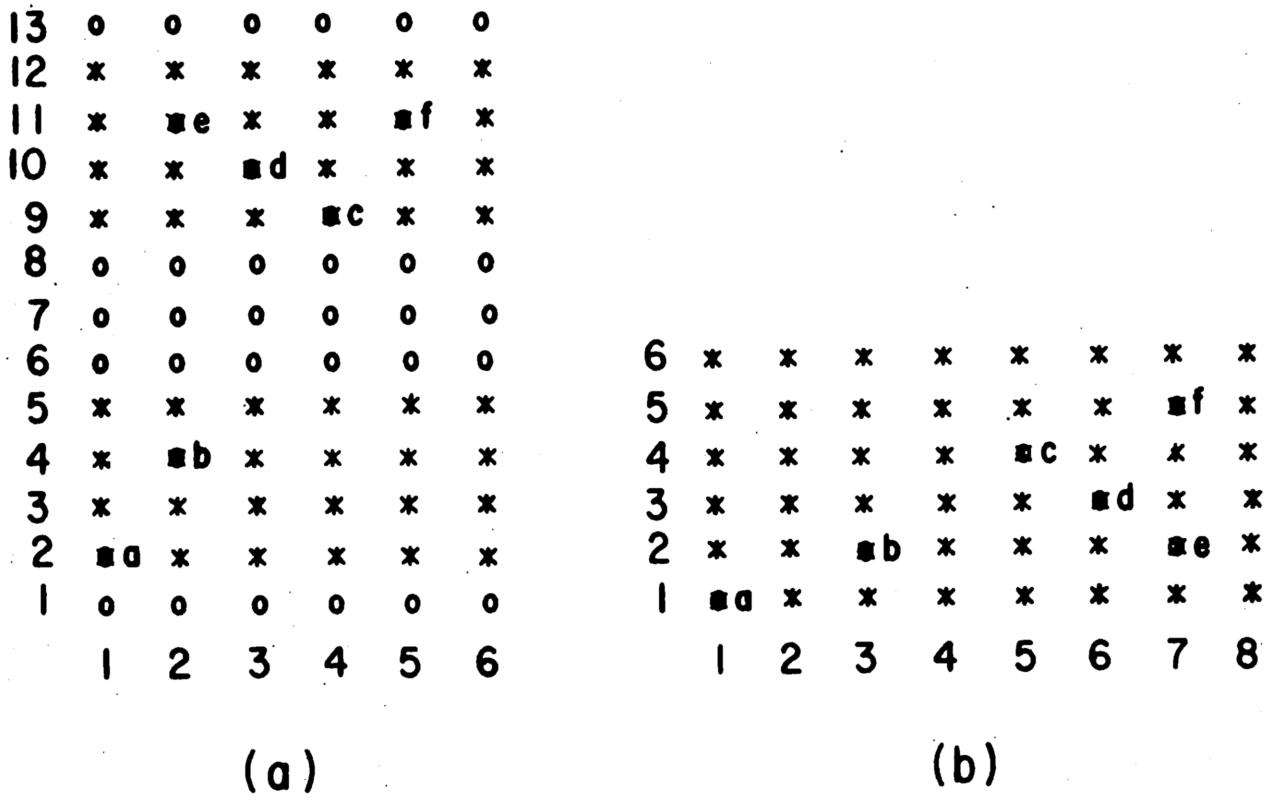
Fig. 2.2



The left-most node of a net on the line

Target of a two-node, step-by-step forward marching process

Fig. 2.3



13	o	o	o	o	o	o
12	x	x	x	x	x	x
11	x	■e	x	x	■f	x
10	x	x	■d	x	x	x
9	x	x	x	■c	x	x
8	o	o	o	o	o	o
7	o	o	o	o	o	o
6	o	o	o	o	o	o
5	x	x	x	x	x	x
4	x	■b	x	x	x	x
3	x	x	x	x	x	x
2	■a	x	x	x	x	x
1	o	o	o	o	o	o
	1	2	3	4	5	6

(a)

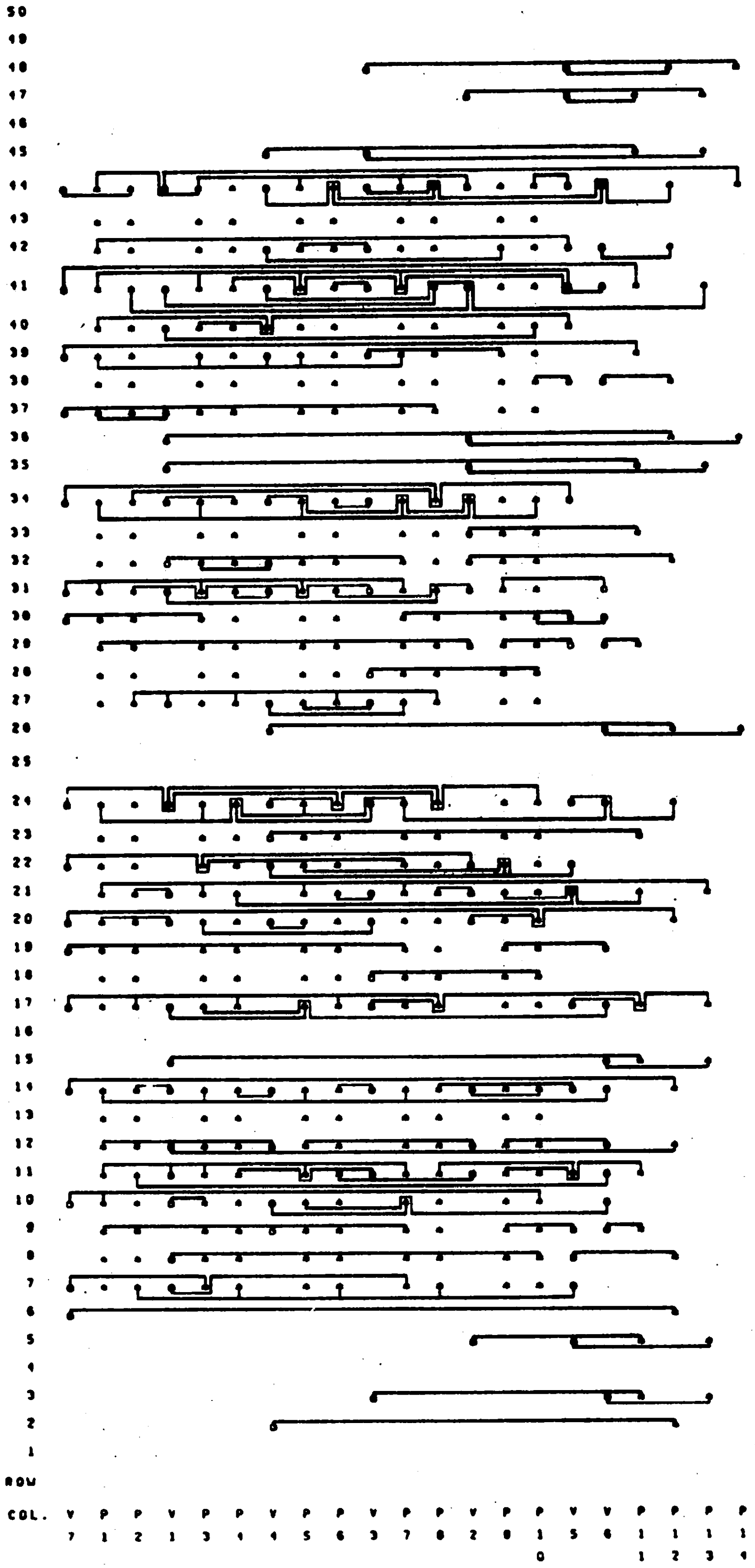
6	x	x	x	x	x	x	x	x
5	x	x	x	x	x	x	■f	x
4	x	x	x	x	■c	x	x	x
3	x	x	x	x	x	■d	x	x
2	x	x	■b	x	x	x	■e	x
1	■a	x	x	x	x	x	x	x
	1	2	3	4	5	6	7	8

(b)

Fig. 2.4

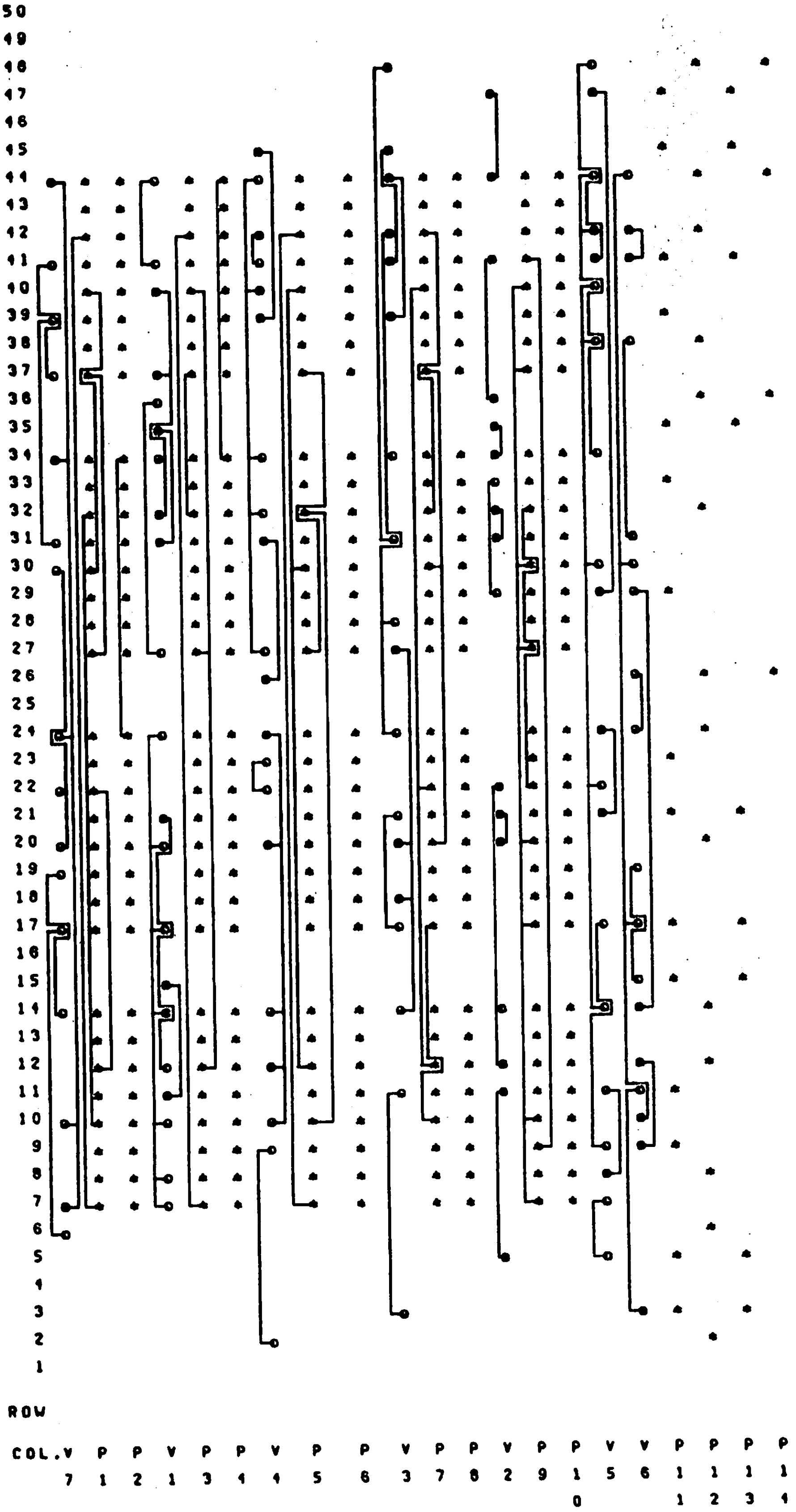
```
12  33 214 6  18       2  29 144 6  51      11  19 921 2  45      11  24 93912  49
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x
50      46    1       40      28    12      35      54    12      23      52    2
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x

 2  33 11410  18       8  29 444 3  51      10  19 521 9  45       9  241039 2  49
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x
50      22    6       40      31     7      35      32    11      23      37   11
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x

 7  33 814 3  18       1  29 64410  51       4  19 321 8  45       4  24 339 7  49
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x
50      26    4       40      15     2      35      34     7      23      48    8
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x

 1  33 614 5  18       6  291044 5  51       9  19102111  45      10  24 53911  49
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x
50      41   10       40      30     1      35      20     2      23      53    9
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x

55  43551327          55  42551617          55  25553617          55  38554347
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x
   1   2      3           4   5      6          7   8      9         10  11     12
 x x x x x x x x        x x x x x x x x      x x x x x x x x       x x x x x x x x

    22  26      27  13      14  15      16  17          18      19  20      21  23      24  25
x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x
  33      40  41      28  29          30      31  32          34      35  36      37  38      39
  x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x
    46  50                  42  43      44                          45          47      48  49
x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x
                                                  51                  52              53      54
  x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x   x
```

Fig. A.1



ROUTING OF A BOARD FROM ILLIAC IV CP UNIT.
HORIZONTAL LAYER, NON-SECTIONED.
NUMBER OF CHIPS= 20.
NUMBER OF SIGNAL NETS= 55.
NUMBER OF VIAS SELECTED= 130.
NUMBER OF TRACKS REQUIRED= 111.
KUH, TING, HORNG (OCTOBER 19, 1976)

Fig. A.2



ROUTING OF A BOARD FROM ILLIAC IV CP UNIT.
VERTICAL LAYER, NON-SECTIONED.
NUMBER OF CHIPS= 20.
NUMBER OF SIGNAL NETS= 55.
NUMBER OF VIAS SELECTED= 130.
NUMBER OF TRACKS REQUIRED= 47.
KUN, TING NORNG (OCTOBER 19, 1976)

Fig. A.3

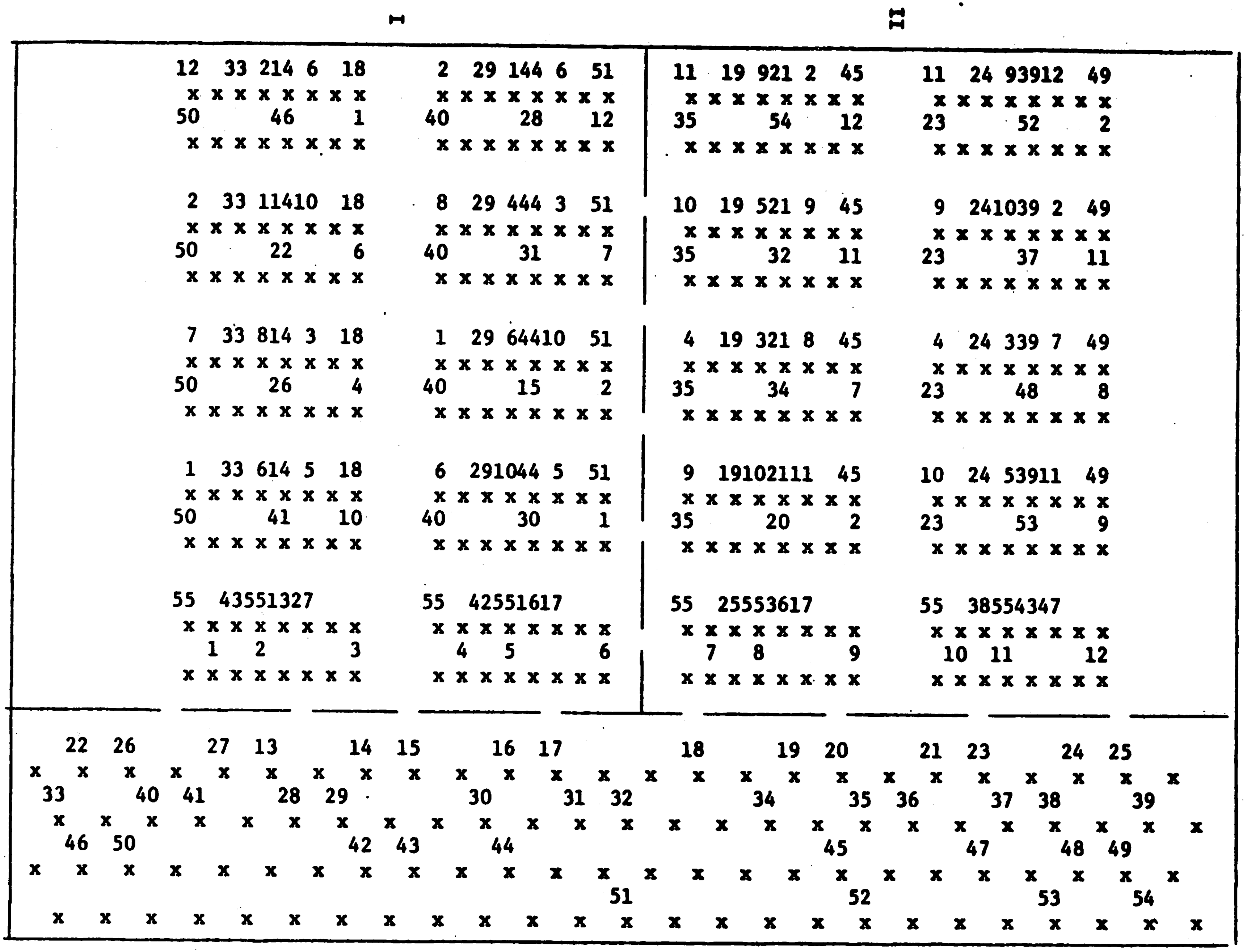
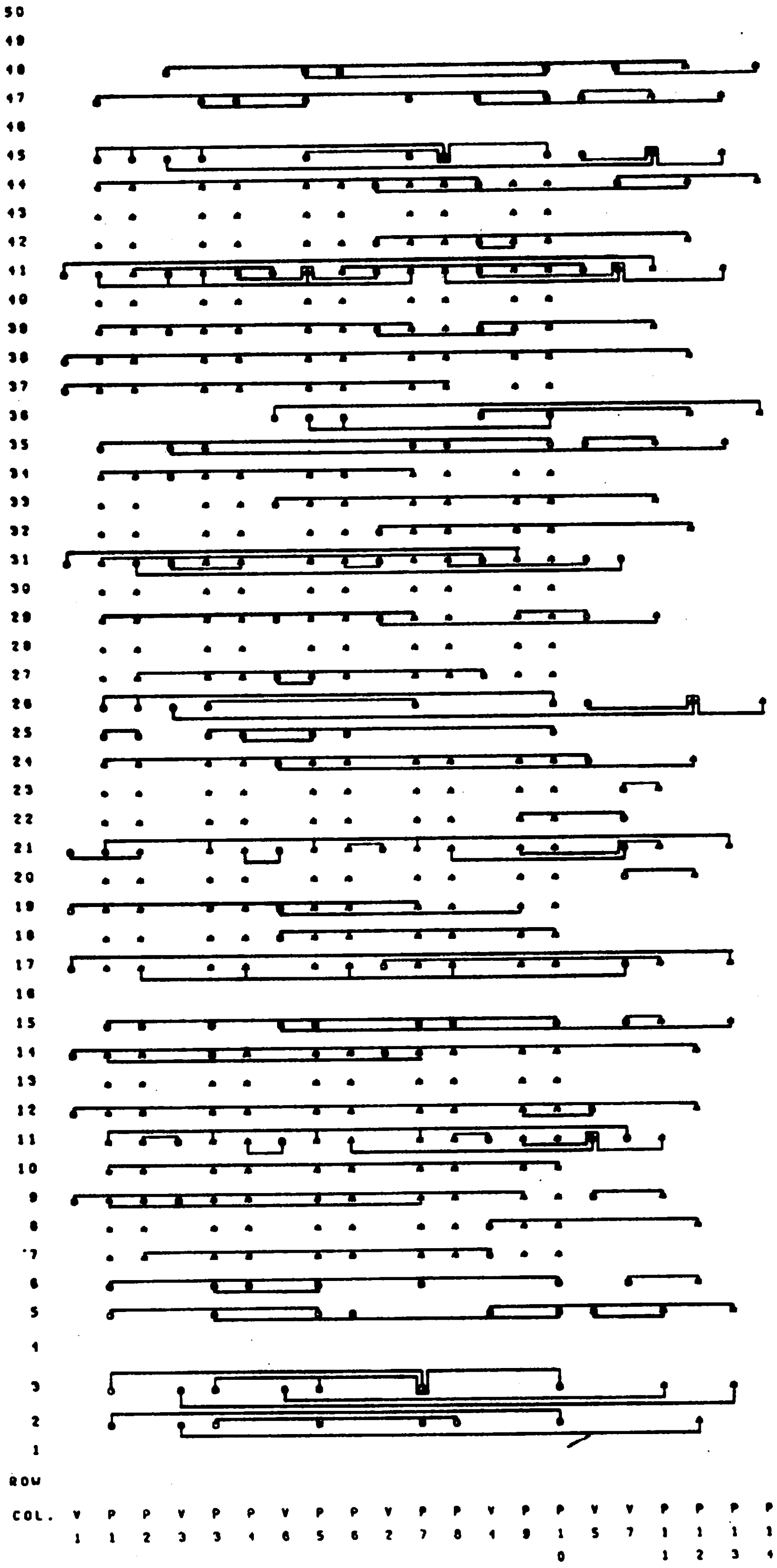
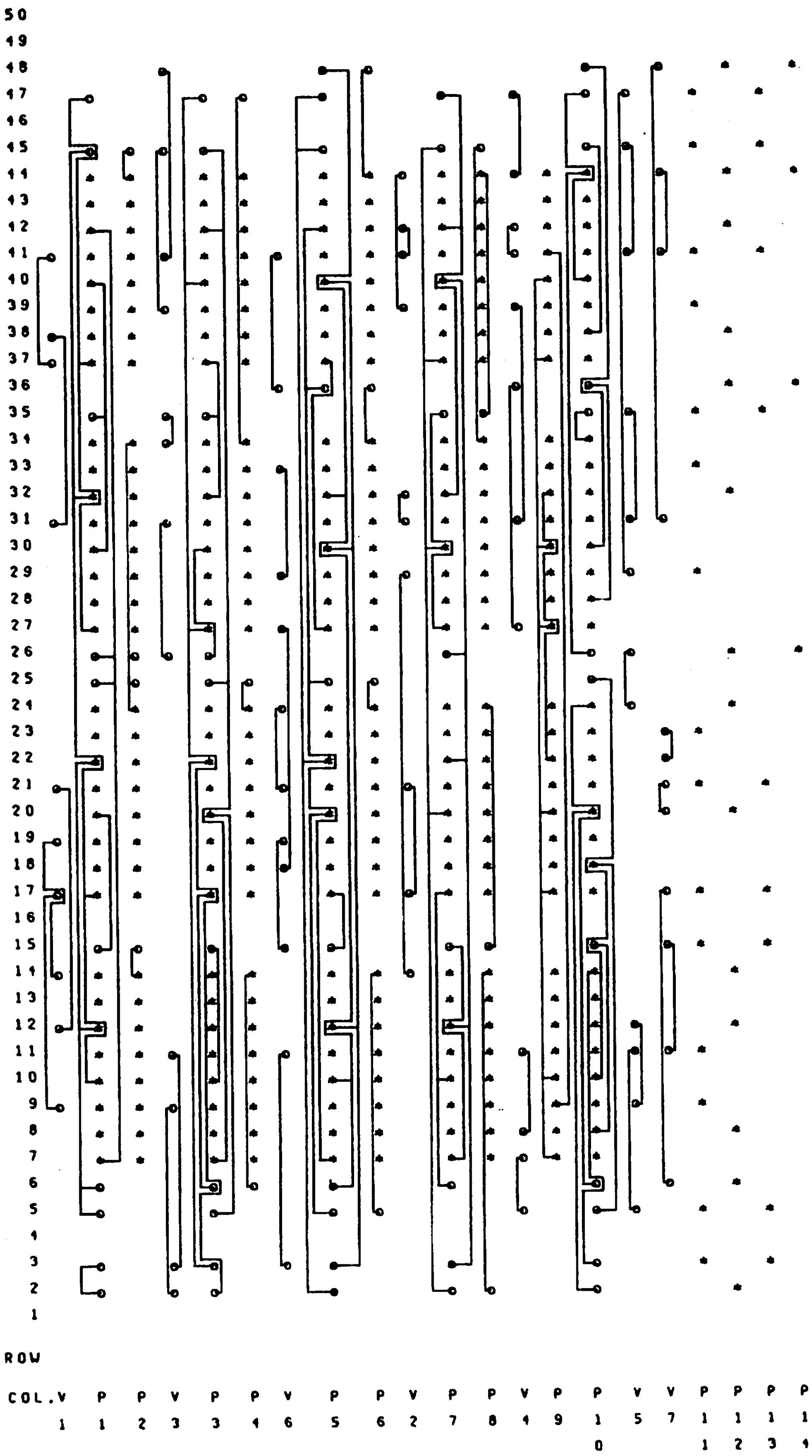

I
II
12 33 214 6 18
50 46 1
2 29 144 6 51
40 28 12
11 19 921 2 45
35 54 12
11 24 93912 49
23 52 2
2 33 11410 18
50 22 6
8 29 444 3 51
40 31 7
10 19 521 9 45
35 32 11
9 241039 2 49
23 37 11
7 33 814 3 18
50 26 4
1 29 64410 51
40 15 2
4 19 321 8 45
35 34 7
4 24 339 7 49
23 48 8
1 33 614 5 18
50 41 10
6 291044 5 51
40 30 1
9 19102111 45
35 20 2
10 24 53911 49
23 53 9
55 43551327
1 2 3
55 42551617
4 5 6
55 25553617
7 8 9
55 38554347
10 11 12
22 26 27 13 14 15 16 17 18 19 20 21 23 24 25
33 40 41 28 29 30 31 32 34 35 36 37 38 39
46 50 42 43 44 45 47 48 49
51 52 53 54

Fig. A.4



ROUTING OF A BOARD FROM ILLIAC IV CP UNIT.
HORIZONTAL LAYER, SECTIONED.
NUMBER OF CHIPS= 20.
NUMBER OF SIGNAL NETS= 55.
NUMBER OF VIAS SELECTED= 145.
NUMBER OF TRACKS REQUIRED= 84.
KUH, TING, HORNG (OCTOBER 20, 1976)

Fig. A.5



ROUTING OF A BOARD FROM ILLIAC IV CP UNIT.
VERTICAL LAYER. SECTIONED.
NUMBER OF CHIPS= 20.
NUMBER OF SIGNAL NETS= 55.
NUMBER OF VIAS SELECTED= 145.
NUMBER OF TRACKS REQUIRED= 56.
KUM. TING. HORNG (OCTOBER 20. 1976)

Fig. A.6

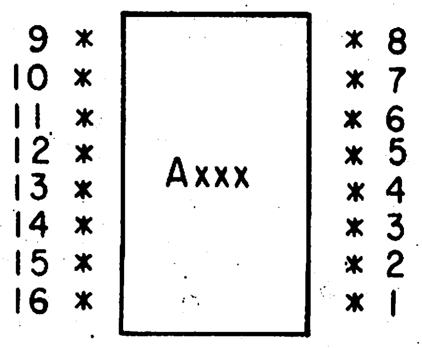


Fig. A.7

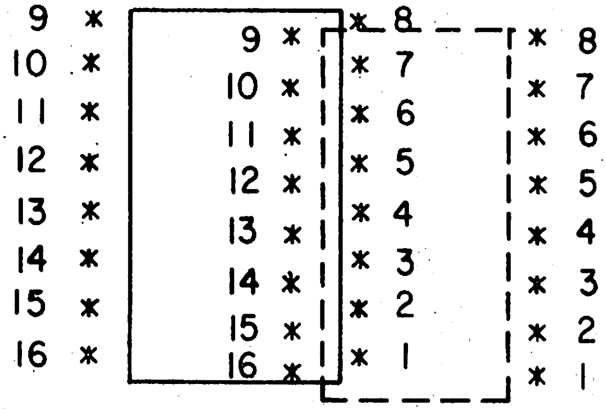


Fig. A.8

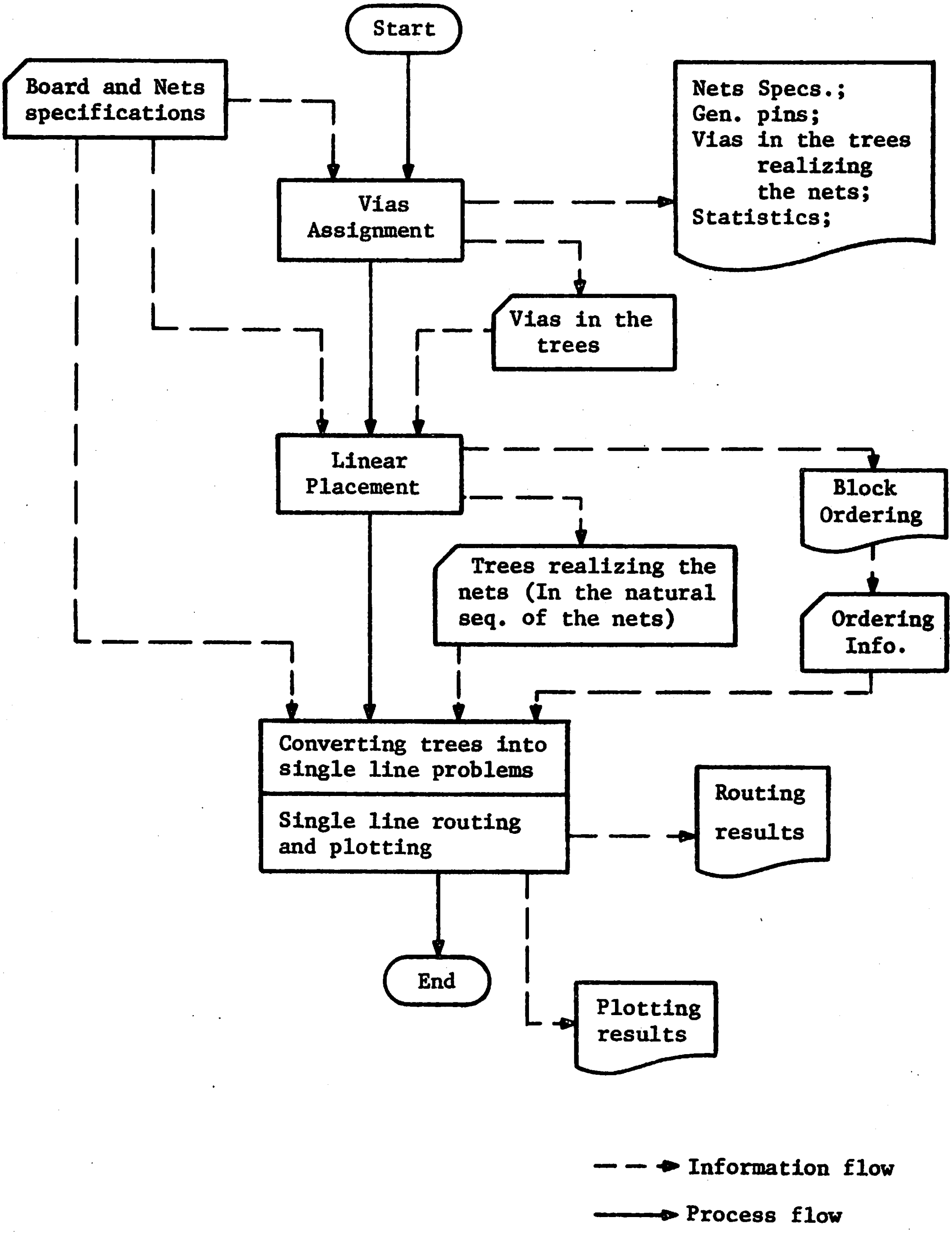

Start
Board and Nets specifications
Vias Assignment
Nets Specs.;
Gen. pins;
Vias in the trees realizing the nets;
Statistics;
Vias in the trees
Linear Placement
Block Ordering
Trees realizing the nets (In the natural seq. of the nets)
Ordering Info.
Converting trees into single line problems
Single line routing and plotting
Routing results
End
Plotting results
Information flow
Process flow

Table 2.1 Process for non-sectioned application

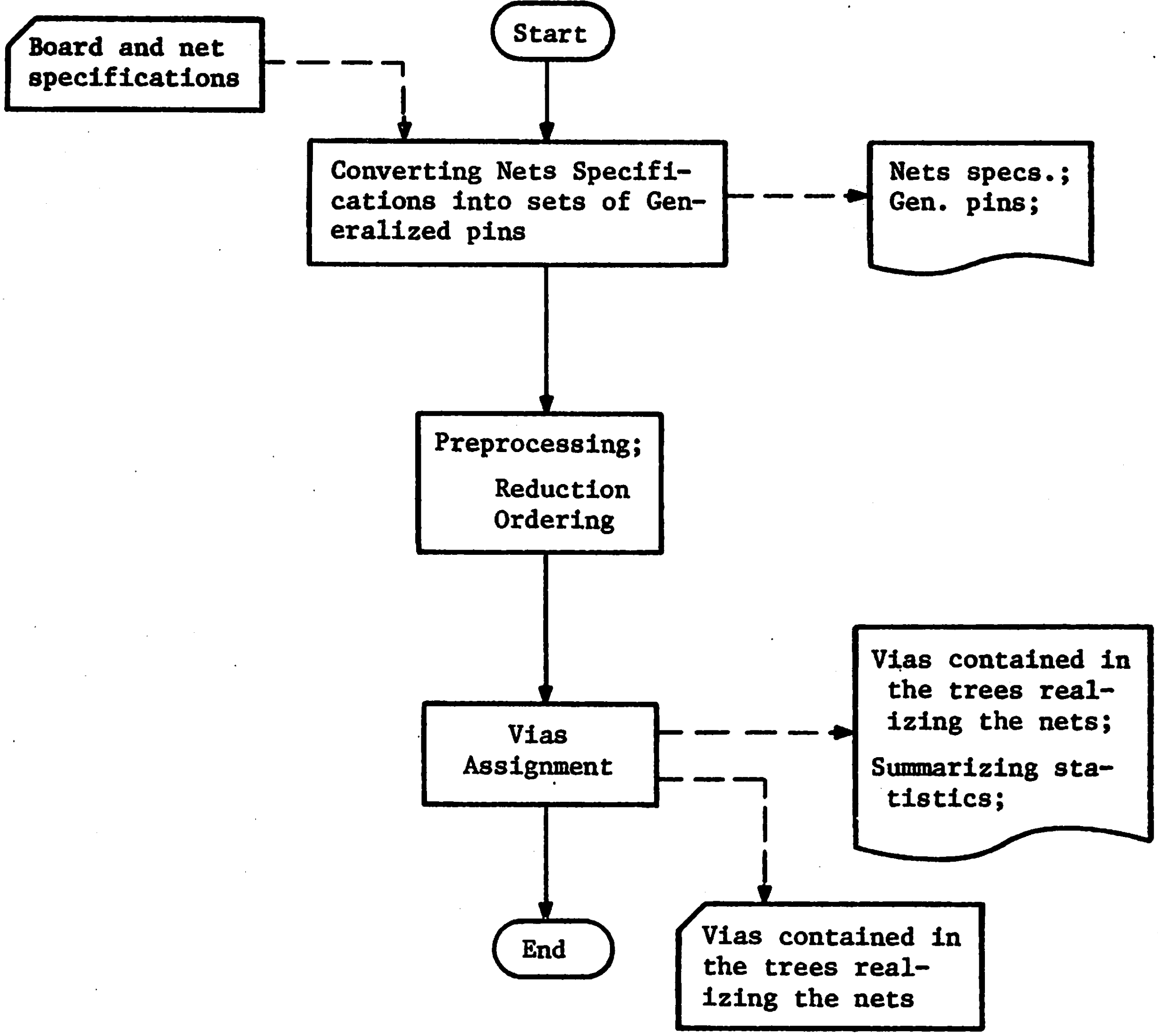
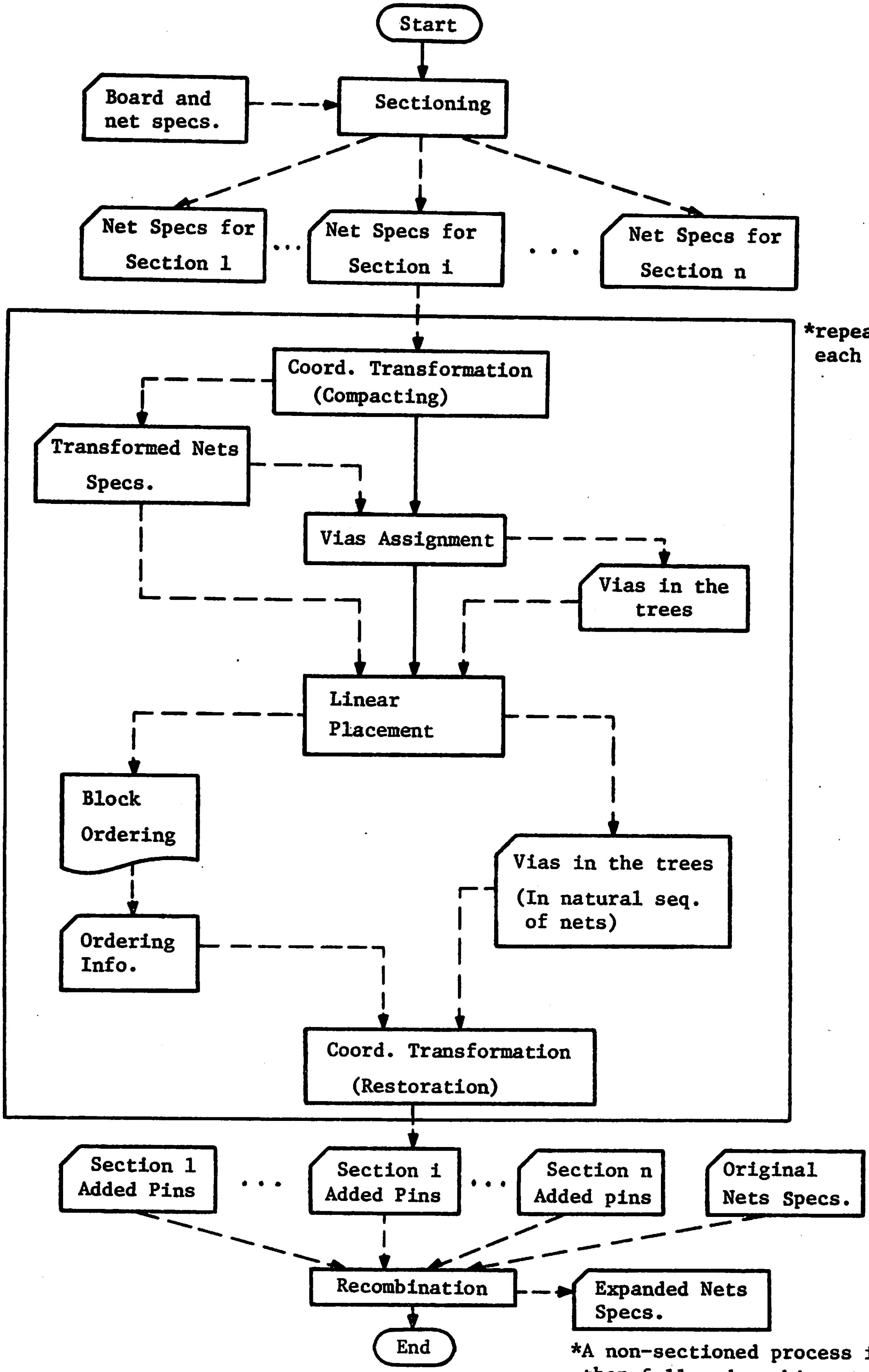

Board and net specifications
Start
Converting Nets Specifications into sets of Generalized pins
Nets specs.; Gen. pins;
Preprocessing;
Reduction
Ordering
Vias Assignment
Vias contained in the trees realizing the nets;
Summarizing statistics;
End
Vias contained in the trees realizing the nets

Table 2.2 Vias assignment phase

Pin row numbers net numbers	1	2	3	4	5	6	7	8	9	10	11	12
1	1		~~2~~^{0}	~~1~~^{0}		2					~~2~~^{0}	
2		1		2	1		3		1		1	
3			3			1		2		1		3
4	1		1		1						1	
5		4	1	2			3		3		4	
6	2	1		1		3			5		4	4
7	3		3		2		1	4	4		1	
8		1	2	1	2	1	2					3
9	1		2	2	1	3		4	5	6		
10		2	1	2	1	2		1				

COUNT	5	5	~~8~~^{7}	~~7~~^{6}	6	6	4	4	5	2	~~6~~^{5}	3

Table 2.3 Example showing the preprocessing operations



*repeated for each section

*A non-sectioned process is then followed, taking this as the original nets specs.

Table 2.4 Process for Sectioned Application

	ILLIAC IV PU BOARD		FIRDB	
	no sectioning	sectioned	no sectioning	sectioned
number of signal nets	55	55	138	138
number of pins on the board	218	218	484	484
number of non-trivial generalized pins	130	130	299	299
number of vias rows	0	12	0	32
number of vias columns	7	7	6	2
number of vias used	130	145	299	306
vias to pins ratio	0.596	0.665	.618	0.632
number of horizontal tracks	111	84	248	152
number of vertical tracks	47	56	125	185

Table 3.1

y-coordinates

6 13	19 26	32 39	45 42	58 65	71 78	84 91	97 104	110 117	123 130	136 143	149 155	162 169	175 182	188 195
			A36		A51	A45		A43	A33	A31	A32			
A35	A34	A37	A50	A49	A48	A41	A42	A44	A17	A38	A29	A30	A40	A47
A21	A18	A19	A20	A8	A24	A23	A16	A28	A25	A46	A39	A27	A26	A22
A0	A1	A10	A4	A15	A9	A5	A12	A13	A11	A3	A2	A6	A7	A14
P4	P3	P15	P1	P9	P5	P6	P14	P2	P8	P13	P7	P10	P12	P11

Table A.1

A035	016	0
A010	007	0
P006	011	0
A037	002	0
A031	002	0
A033	002	0
A029	002	0
A042	002	0
A038	002	0
A039	002	0
A034	002	0
A043	004	0
A045	002	0
A024	002	0
A026	002	0
A021	002	0
A027	002	0
A018	002	0
A038	016	1
A048	014	1
A012	009	1
P015	004	1
P010	016	1
P009	007	1
P013	003	1
P002	016	1
P013	015	1
P013	004	1
P003	015	1
A013	016	1
P001	007	1
P009	015	1
P007	003	1
P004	004	1
P015	005	1
P007	008	1
A048	016	0
P008	016	0
P008	007	0
A036	002	0
A030	002	0
A032	002	0
A040	002	0
A041	002	0
A246	002	0
A035	002	0
A013	001	0
A044	004	0
A022	002	0
A025	002	0
A023	002	0
A020	002	0
A019	002	0
A017	002	0
P010	007	1
A011	009	1
A011	001	1
P001	008	1
P012	011	1
P010	004	1
P012	007	1
P006	012	1
P013	011	1
P003	003	1
P002	015	1
A013	014	1
P011	004	1
P008	008	1
P010	003	1
P005	011	1
P006	015	1
P008	003	1

Table A.2 (continued)

A030	009	0	A031	009	0	A032	009	0	A033	009	0
A017	011	1	A047	001	0	A038	009	0	A039	009	0
A040	009	0	A023	011	0	A024	011	0	A025	011	0
A026	011	1	A047	008	0	A018	011	0	A019	011	0
A020	011	0	A021	011	0	A034	009	0	A035	009	0
A036	009	0	A037	009	1	A047	002	0	A041	009	0
A042	009	0	A043	009	0	A044	009	1	A047	009	0
A022	011	1	A047	004	0	P011	008	1	A007	001	0
A047	014	1	A007	002	0	A047	011	1	A048	005	0
A046	011	0	A029	011	0	A034	011	0	A035	011	0
A036	011	0	A037	011	0	A043	011	0	A045	012	1
A048	007	0	A027	012	0	A030	011	0	A031	011	0
A032	011	0	A033	011	0	A017	012	1	A048	008	0
A018	012	0	A019	012	0	A020	012	0	A021	012	0
A041	011	0	A042	011	1	A048	009	0	A038	011	0
A039	011	0	A040	011	0	A022	012	0	A023	012	0
A024	012	0	A025	012	0	A026	012	1	A008	008	0
P009	016	1	A008	007	0	A048	011	1	A049	005	0
A027	013	0	A030	012	0	A031	012	0	A032	012	0
A033	012	0	A017	013	1	A049	007	0	A046	012	0
A029	012	0	A022	013	0	A023	013	0	A024	013	0
A025	013	0	A026	013	1	A049	001	0	A018	013	0
A019	013	0	A020	013	0	A021	013	0	A034	012	0
A035	012	0	A036	012	0	A037	012	1	A049	008	0
A041	012	0	A042	012	1	A049	002	0	A038	012	0
A039	009	0	P009	004	1	A008	001	0	A049	014	1
A008	002	0	A049	011	1	A050	005	0	A046	013	0
A029	013	0	A017	014	0	A018	014	0	A019	014	0
A020	014	0	A021	014	1	A050	007	0	A027	014	0
A030	013	0	A031	013	0	A032	013	0	A033	013	1
A050	001	0	A023	014	0	A041	013	0	A042	013	1
A050	008	0	A034	013	0	A035	013	0	A036	013	0
A037	013	0	A043	012	0	A044	012	0	A045	011	0
A045	014	1	A050	009	0	A038	013	0	A039	013	0
A040	013	0	A022	014	0	A024	014	0	A025	014	0
A026	014	1	A050	004	0	P007	004	1	A009	008	0
A050	014	1	A009	007	0	A050	011	1	A051	005	0
A029	014	0	A030	014	0	A032	014	1	A051	007	0
A027	016	0	A046	014	0	A031	014	0	A033	014	0
A017	016	0	A020	016	0	A034	014	0	A037	014	1
A051	001	0	A018	016	0	A019	016	0	A021	016	0
A035	014	0	A037	014	1	A051	008	0	A038	014	0
A039	014	0	A040	014	0	A022	016	0	A026	016	0
A041	014	0	A042	014	1	A051	002	0	A023	016	0
A024	016	0	A025	016	1	A051	004	0	P006	007	1
A009	001	0	A051	014	1	A009	002	0	A051	011	1
A048	004	0	A017	001	0	A019	001	0	A020	001	0
A036	016	0	A037	016	0	A039	016	0	A040	016	1
A048	001	0	A027	001	0	A046	016	0	A029	016	0
A030	016	0	A031	016	0	A032	016	0	A033	016	1
A048	002	0	A018	001	0	A021	001	0	A034	016	0

Table A.2 (continued)

FIRDB

P004	003	0	A000	016	1	P004	007	0	A000	014	1
A000	002	0	A001	016	1	A002	008	0	A003	009	0
A003	016	0	A004	001	0	A004	009	0	A005	009	0
A005	014	1	A003	004	0	A006	011	0	A007	011	1
A003	005	0	A006	013	0	A007	013	1	A005	004	0
A008	011	0	A009	011	1	A005	005	0	A008	013	0
A009	013	1	A004	004	0	A010	011	1	A004	005	0
A010	013	1	A001	001	0	A003	012	0	A003	013	0
A005	012	0	A004	013	1	A001	002	0	A004	012	0
A004	013	1	P007	007	0	A002	011	1	A011	002	0
A003	008	0	A004	011	0	A005	008	1	A012	007	0
A003	001	0	A004	014	0	A005	011	1	P007	004	0
A002	012	1	P006	016	0	A012	013	1	A013	004	0
A011	012	1	P011	011	0	A014	009	1	A014	008	0
A006	012	1	P011	007	0	A014	012	1	A014	005	0
A006	014	1	P011	003	0	A014	013	1	A014	004	0
A007	012	1	P010	015	0	A014	016	1	A014	001	0
A007	014	1	P009	015	0	A015	009	1	A015	008	0
A008	012	1	P009	016	0	A015	012	1	A015	005	0
A008	014	1	P009	011	0	A015	013	1	A015	004	0
A009	012	1	P009	012	0	A015	016	1	A015	001	1
A009	014	1	P007	003	0	A002	016	1	A002	002	0
A010	012	1	A002	005	0	A013	013	0	A012	011	1
A012	004	0	A013	012	0	A012	012	1	P006	015	0
A012	008	1	P009	003	0	A011	008	1	P008	015	0
A011	011	1	A016	005	0	A017	007	0	A018	007	0
A019	007	1	A016	007	0	A020	007	0	A021	007	1
A016	008	0	A022	007	0	A023	007	0	A024	007	0
A025	007	0	A026	007	1	A016	009	0	P014	007	1
A006	007	0	A016	011	0	A027	007	1	A028	005	0
A029	007	0	A030	007	0	A031	007	0	A032	007	0
A033	007	1	A028	007	0	A017	008	0	A018	008	0
A019	008	0	A020	008	0	A021	008	1	A028	001	0
A034	007	0	A035	007	0	A036	007	0	A037	007	0
A038	007	0	A039	007	0	A040	007	1	A028	008	0
A022	008	0	A023	008	0	A024	008	0	A025	008	0
A026	008	1	A028	002	0	A041	007	0	A042	007	1
A028	009	0	A043	007	0	A044	007	0	A045	008	0
A045	013	1	A028	004	0	P013	007	1	A006	001	0
A028	014	0	A027	008	0	A046	007	1	A006	002	0
A028	011	1	A016	004	0	A046	008	0	A032	008	0
A033	008	0	A038	008	0	A040	008	0	A023	009	0
A025	009	0	A026	009	1	A016	001	0	A027	009	0
A029	008	0	A030	008	0	A031	008	0	A017	009	1
A016	002	0	A018	009	0	A019	009	0	A020	009	0
A021	009	0	A034	008	0	A035	008	0	A036	008	0
A037	008	1	A016	016	0	A039	008	0	A022	009	0
A024	009	0	A042	008	0	A043	008	0	A044	008	0
A045	009	0	A045	016	1	A007	008	0	A041	008	0
P006	011	1	A007	007	0	A016	014	1	A047	005	0
A027	011	0	A029	009	1	A047	007	0	A046	009	0

Table A.2