A NEW APPROACH TO ROUTING OF TWO-LAYER PRINTED CIRCUIT BOARD

by

M. Marek-Sadowska and E. S. Kuh

Memorandum NO. UCB/ERL M80/30

14 July 1980

A NEW APPROACH TO ROUTING OF TWO-LAYER PRINTED CIRCUIT BOARD

by

M. Marek-Sadowska and E. S. Kuh

Memorandum No. UCB/ERL M80/30

14 July 1980

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

A NEW APPROACH TO ROUTING OF TWO-LAYER PRINTED CIRCUIT BOARD

M. Marek-Sadowska* and E. S. Kuh

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

ABSTRACT

This paper presents a new algorithm for routing two-layer printed circuit boards. Circuit components are mounted on top of the board, and conductor wires are to be layed out on the board such that circuit connections can be properly made. The proposed approach gives 100% routability. It depends on a specific via assignment and definite routing rules for the two layers. Computer program based on the presented algorithm was written. Its implementation is presented along with testing examples.

*On leave from Institute of Electron Technology, Technical University of Warsaw, ul. Koszyleowa 75, 00-665 Warsaw, Poland

Introduction

This paper considers a new routing algorithm. It is assumed that a board with modules mounted on its top is given. The problem considered here is how to interconnect terminals of the modules according to specifications by means of printed conductors which are layed on a two-sided board.

It is assumed that the board has fixed geometries, i.e. it has fixed plated-through holes, uniformly spaced on a rectangular grid. Conductor pins (drilled-through holes to reach both layers) and vias (plated-through holes to be used for interconnections between layers) alternate on each row. It is illustrated in Fig. 1. In absence of any constraints, routability is of no problem, and it can always be completed on one layer. However, when constraints are imposed it is necessary to use multilayers. In such case conductor paths have to be routed through vias to reach from one layer to another. The constraints introduced here deal with shape of conductor paths; proposed algorithm gives 100% routability for these feasible shapes of connections, when the two layers are available.

Formulation of the Problem

The problem is then to interconnect each set of the pins to be made electrically common by means of conductors paths laid on each layer together with vias. Following constraints are imposed on the shape of interconnections.

First, only rectilinear paths are allowed, i.e., paths contain only horizontal and vertical conductor segments. Second, in the upper layer only upward and downward zigzagging is allowed [Fig. 2]. Third, in the bottom layer upward and downward zigzagging is allowed with some modifications: forward and backward zigzagging is not allowed between two or

more consecutive columns [Fig. 3].

The problem is stated as follows. Given a net list

$$L = \{N_1, N_2, \ldots, N_k\} \text{ where } N_i = \{P_{a_1 b_1}, P_{a_2 b_2}, \ldots, P_{a_k b_k}\}.$$

Find how to layout paths on each layer such that the above conditions are satisfied and paths that belong to different nets do not intersect on the same layer. It is assumed that between two consecutive rows or columns there is enough space to place as many connections as necessary to fulfill the above requirements. However, since this assumption is not practical, the algorithm is constructed in such way as to minimize the number of paths to be layout between two consecutive rows or columns.

Basic Approach

The design of interconnections is divided into three interrelated phases:

a. decomposition of interconnections into portions of each layer

b. via assignment

c. layout the wire pattern on each layer.

The approach adopted here is as follows:

1. Each multi-pin net is partititioned into two-pin subnets. If all the subnets constituting given net are connected then the net itself is connected. The partition of net is done in such way that if two pins are assigned to the same subnet, if and only if they belong to the nearest columns, that is, there exists no forward and backward zigzagging. See example in Fig. 4.

2. Decomposition of interconnections into portions placed on separate layers applies following rules:

a. at least one via is assigned per each subnet

b. if pins $P_{\alpha\beta}$ and $P_{\gamma\delta}$ form a subnet then this one via assigned to the subnet is in the via column placed immediately before pin column number δ. [Fig. 5].

3. In the upper layer are placed essentially horizontal connections, in the bottom layer essentially vertical connections.

In the basic approach only one via per subnet is allowed. Going from left to right, we follow the rule that connections from pin to via are layed out in the upper layer and from via to pin in the bottom layer.

The above scheme gives 100% routability for any net list. It can be proven as follows. For each subnet there always exists at least one via that can be assigned to it. It is so because if we consider a board consisting of r rows and k columns of pins then in each pin column no more than r subnets can terminate so there will be a via for each of them. The algorithm given below guarantees that wires on each layer will not intersect.

Suppose that in the upper layer all the connections up to the pin column number L are already made. The simplest algorithm to generate connections between pin columns number L and L + 1 is as follows:

Step 1: Consider subnets from top down to consecutive order. A subnet either has reached column number L from the left or originates in column number L. It can either terminate in pin column number L + 1 or not.

Step 2: If the actually checked subnet terminates in the pin column L + 1 then assign to it the first unoccupied via from the via column placed immediately before the pin column L + 1 top down; go to step 3. If the connection does not terminate in column L + 1 go to step 3.

Step 3: If there are any interconnections left that were not examined already, go to step 1; otherwise follow step 4.

Step 4: Make the connections to the via column. Shape of these

connections is forced by connections that terminate in the nearest via column.

Step 5: Extend the nonterminating connections to the next pin column.

Step 6: End of the algorithm.

An example of this is shown in Fig. 6(a) and Fig. 6(b).

Now, consider interconnections to be placed in the bottom layer. These are connections between two consecutive via and pin columns; via column is on the left hand side, pin column on the right hand side.

The algorithm to complete interconnections in this layer is as follows:

Step 1: Take the consecutive occupied via from the top. Let it correspond to subnet number i. It is to be connected to pin i on the right.

Step 2: Determine how far down on the via column this connection has to go. The path has to be below all vias having their corresponding pins placed above pin i.

Step 3: Generate the connection in such a way that when it goes down, the path either stays on the right or left side of the via column. If it passes unoccupied vias or those vias that have their corresponding pins below pin i the path is on the right hand side. Otherwise it is on the left hand side.

Step 4: If there are any vias left, go to step 1; otherwise follow step 5.

Step 5: End of the algorithm.

Fig. 6(c) shows an example how connections are made on the bottom layer.

Fig. 7 shows the routing of a simple printed circuit board using this simplest version of the algorithm.

Now, let us consider a printed circuit board of r rows and k columns. It contains r x (k-1) vias. If there are r nets, each of them having pins in every column then all vias will be occupied. In all other situations some vias will be left unused.

Modifications of the Basic Approach

The routing obtained by application of the basic algorithm can be improved by taking advantage of unused areas in the upper layer, or unoccupied vias or both. In practice there are some additional requirements that are to be satisfied by routing. One of them considers the number of parallel conductor segments that can be layed in vertical and horizontal channels. It is either stated in a form that the number of parallel tracks has to be minimized or that it is limited up to a certain number. The other requirement is to minimize the total length of connections. The modifications, which are possible in most of the practical cases, can be used to improve routing that would be obtained by application of the basic algorithm so that channel density, or total length of connections, or both are decreased. Of course it is obvious that just by adding these improvements to the basic algorithm the global minimum of channel density or connections length will not be achieved.

The use of empty vias will be discussed first. They can be used in two ways. First it is not necessarily to assign consecutive vias from the top to terminating subnets. To illustrate this case let us consider again the example of Fig. 6(b). Here, in the upper layer the maximal density in horizontal channel is 4, in vertical channel is 6.
But in the via column there are 3 vias unoccupied. If vias are assigned to terminating nets in such a way as in Fig. 8a, the maximal density in horizontal channels is reduced to 3, in vertical to 1.

After applying this modification still only one via is assigned per subnet. If there were p empty vias in the considered via column, there are still p empty vias after the modification. If $p \geq 2$ then these empty vias can be assigned to connections passing in their intermediate neighborhood to transfer them from highly occupied channels to those which are less. This technique is illustrated in Fig. 9 and its application to the previous example is shown in Fig. 8b.

The technique of transferring connections is applied in first place if in a channel there are more than two subnets terminating in the same pin column. Such connections are moved to other channels if it is possible. The reason of doing this is that if more than 2 subnets from one horizontal channel terminate in the same pin column then it always causes increase of density in vertical channels and sometime also in horizontal channels. This situation is illustrated in Fig. 10.

The second modification concerns transferring interconnections from bottom layer to upper layer if there is space to do it. It is the only modification that leads to minimization of channel densities in this layer. An example is shown in Fig. 11.

Fig. 12a shows example of the same printed circuit board as in Fig. 7 but routed by algorithm using the above modifications. In Fig. 12b is shown the same printed circuit board routed by conventional method [2].

Implementation

Computer program based on above algorithm was written. It applies the basic algorithm together with the modification described above. This program is constructed in such a way that it performs only local analysis while completing connections. It does not allow back tracking or changes in already drawn connections. The major criterion applied

was to obtain the smallest possible channel density. The other one was to avoid unnecessary bending of wires. The program does not guarantee the optimal solution.

In each step of analysis the program generates 2 vectors describing the sequence of nets passing through pin and via column. An example is shown in Fig. 13. It is sufficient to use these vectors to draw connections in the upper layer between pin and via columns or via and pin columns in the lower layer. Length of these vectors is equal to the number of passing nets + number of rows of the PCB. Similar vectors are generated for the bottom layer (their dimension is only r (number of rows)).

Each step of the algorithm up to generation of vectors that describe routing is linear as a function of number of nets passing through the procedure is a linear function of $k(r+\Sigma$ all densities in the upper layer) + $(k-1)r$. It is so because analysis is performed for consecutive pin columns. For each column each connection passing through or originating in it is checked once. Also in each via column unused vias are to be found. The only nonlinear process is how to recover the actual shape of the wires from these vectors. Computational complexity is here of order $(k-1)r^2 + (k-1)(\text{maximal density} + r)^2$. Quadratic terms are due here to the fact that while drawing one graphic terminal each actually processed interconnection requires information about shapes of all other.

Conclusions

Major advantages of this algorithm are:

(1) 100% routability

(2) linearity

Major disadvantages:

(1) Due to local analysis channel density in the bottom layer is unpredictable and it can be reduced only if some connections can be transferred from there to the upper layer.

(2) Also in the upper layer it can happen that local analysis is not sufficient to avoid crowding of wires in some channels.

Two examples of routing completed by this program are shown in Figs. 14 (a) and (b).

Acknowledgement

Research sponsored by the National Science Foundation Grant ENG-78-24425.

Literature

[1] B. S. Ting, E. S. Kuh, and I. Shirakawa,"The Multilayer Routing Problem : Algorithms and Necessary and Sufficient Conditions for the Single-Row Single-Layer Case," IEEE Transactions on Circuits and Systems, Vol. CAS-23, No. 12, December 1976, pp. 768-778.

[2] S. Tsukiyama, I. Shirakawa, and S. Asahama,"An Algorithm for the Via Assignment Problem in Multilayer Backboard Wiring," IEEE Transactions on Circuits and Systems, Vol. CAS-26, No. 6, June 1979.

[3] B. S. Ting, and E. S. Kuh,"An Approach to the Routing of Multilayer Printed Circuit Boards," Proc. IEEE Int. Symp. on Circuits and Systems, 1978, pp. 902-911.

[4] H. C. So, "Some Theoretical Results on the Routing of Multilayer Printed Wiring Boards," Proc. IEEE Int. Symp. on Circuits and Systems, 1974, pp. 206-303.

Figure Captions

Fig. 1. Geometry of the printed circuit board. x denotes pin, o-via. There is one via column less than pin columns.

Fig. 2. Upper layer.

(a) Only upward and downward zigzagging is allowed in the upper layer.

(b) Backward and forward zigzagging is not allowed in the upper layer.

Fig. 3. Bottome layer.

(a) Upward and downward zigzagging in allowed. Backward and forward zigzagging is allowed only around via column.

(b) Backward and forward zigzagging is not allowed around pin column.

(c) Backward and forward zigzagging is not allowed around two or more columns.

Fig. 4. Decomposition of net $N_1 = \{p_{51}, p_{12}, p_{44}, p_{16}\}$ into three subnets:

$$N_1^{(1)} = \{p_{51}, p_{12}\}, N_1^{(2)} = \{p_{12}, p_{44}\}, N_1^{(3)} = \{p_{44}, p_{16}\}.$$

Other partitions are not allowed. Example of not allowed partition is:

$$N_1^{(1)} = \{p_{51}, p_{44}\}, N_1^{(2)} = \{p_{12}, p_{16}\}, N_1^{(3)} = \{p_{12}, p_{44}\}.$$

Fig. 5. Via assigned to the subnet $\{p_{\alpha\beta}, p_{\gamma\delta}\}$ is in the via column placed immediately before pin column number δ.

Fig. 6. Connections between two consecutive pin columns generated by the simplest algorithm.

(a) All interconnections are completed up to pin column number ℓ.

(b) Interconnections layed out in the upper layer.

(c) Connections in the bottom layer.

Fig. 7. Routing completed by the simplest version of the algorithm.

Fig. 8. Via assignment modifications.

(a) First via assignment modification applied.

(b) Second via assignment modification applied.

Fig. 9. Additional via assignment technique.

(a) Connection "a" has to be transferred from horizontal channel number 3 to channel 5.

(b) Vias in third and fifth row are additionally assigned to connection "a". Connection between vias 3 and 5 is placed in the bottom layer.

Fig. 10. More than two subnets from one horizontal channel terminate in the same pin column.

Fig. 11. Technique of transferring connections from the bottom layer to the upper layer.

(a) Initial routing without modifications.

(b) Connection number 1 transferred from the bottom layer to upper layer.

Fig. 12. Routing of the same PCB's obtained by two different methods.

(a) Connections routed by the modified version of the algorithm. Length of connections = 34, 5 via holes left.

(b) Connections routed by conventional method [2]. Length of connections = 38. All via holes used.

Fig. 13. W1 described situation in pin column, - sign denotes net originating in the column, 0 denotes empty pin.

W2 describes situtation in via column, - sign denotes net terminating in this column, 0 denotes empty via.

Fig. 14. An example with 18 multiplier nets randomly generated: (a) using simple algorithm, and (b) with modifications.

X O X O X O X

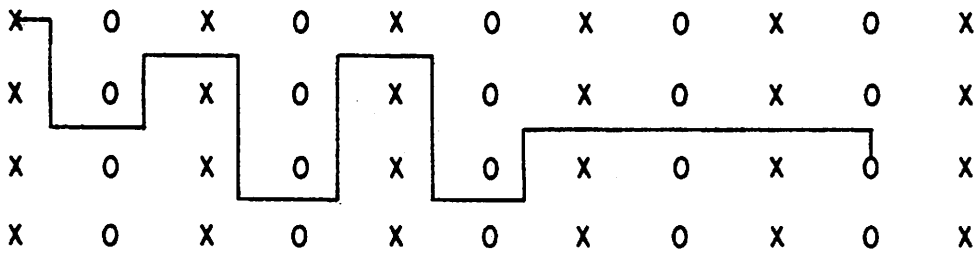
X O X O X O X

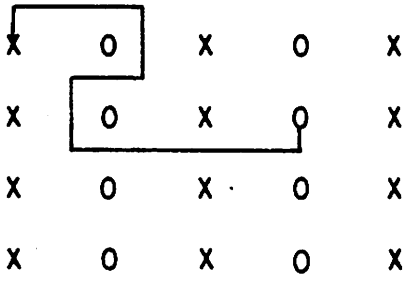
X O X O X O X

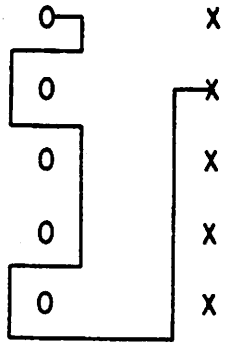
X O X O X O X

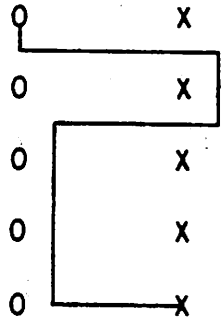
X O X O X O X

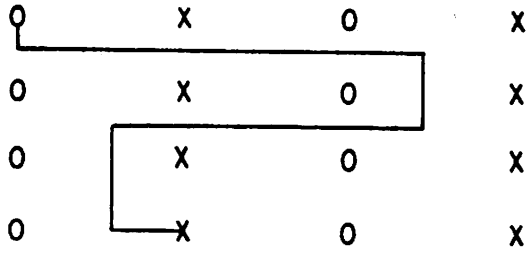
X O X O X O X


X O X O X O X O X O X
X O X O X O X O X O X
X O X O X O X O X O X
X O X O X O X O X O X

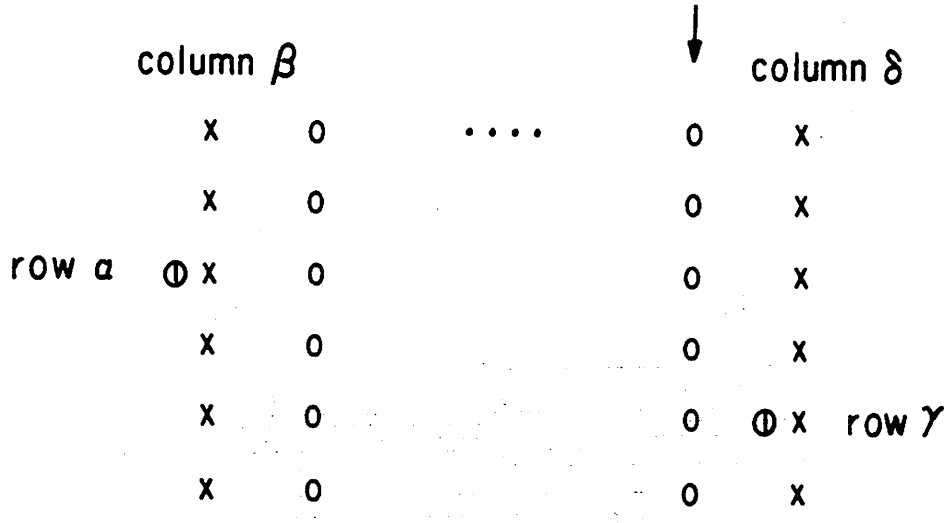

X O X O X
X O X O X
X O X O X
X O X O X

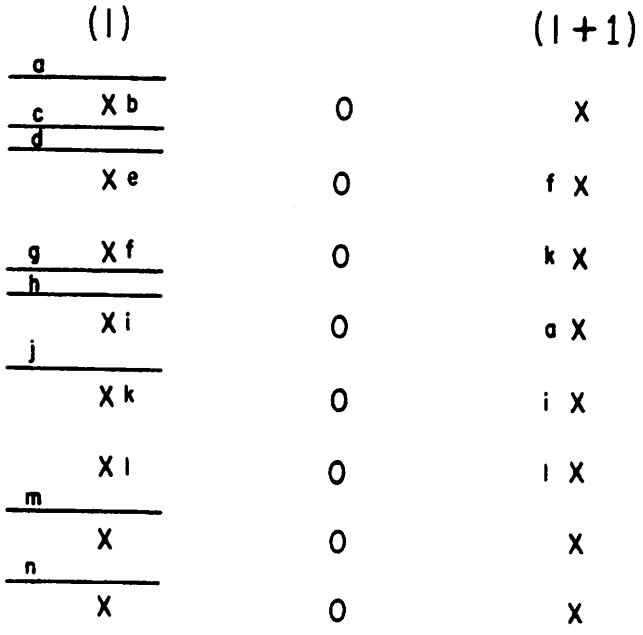

O X
O X
O X
O X
O X

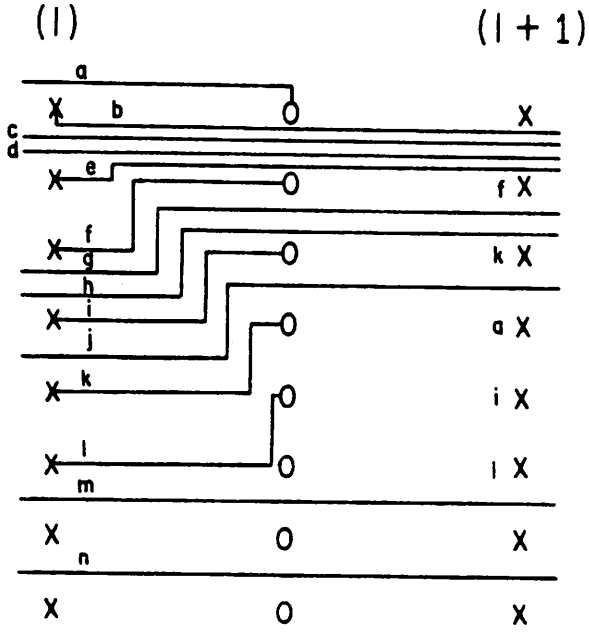

O X
O X
O X
O X
O X

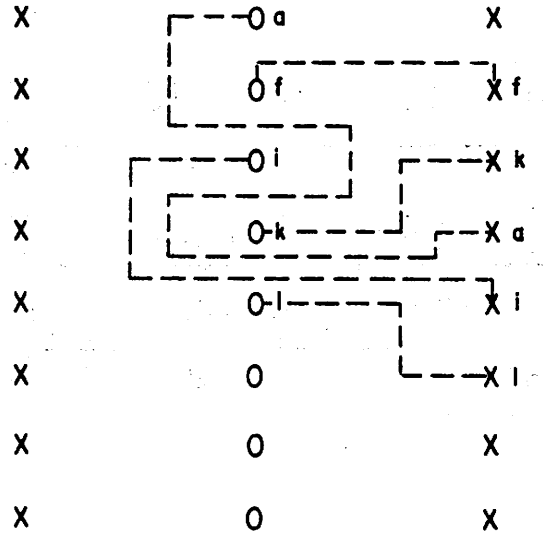

O X O X
O X O X
O X O X
O X O X

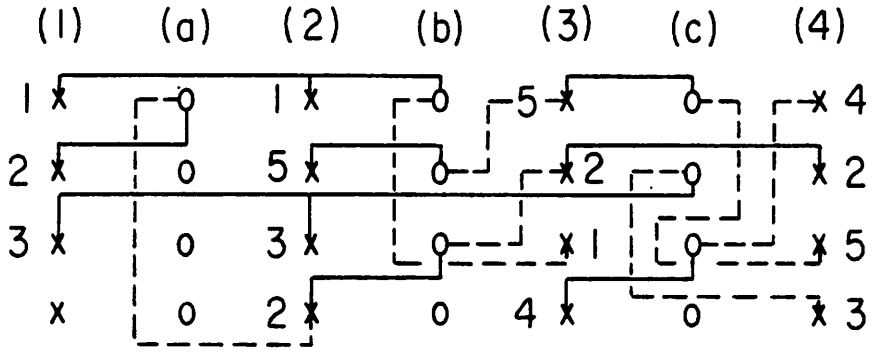
	1		2		3		4		5		6
1	x	o	ⓘx	o	x	o	x	o	x	o	xⓘ
2	x	o	x	o	x	o	x	o	x	o	x
3	x	o	x	o	x	o	x	o	x	o	x
4	x	o	x	o	x	o	xⓘ	o	x	o	x
5	ⓘx	o	x	o	x	o	x	o	x	o	x
6	x	o	x	o	x	o	x	o	x	o	x

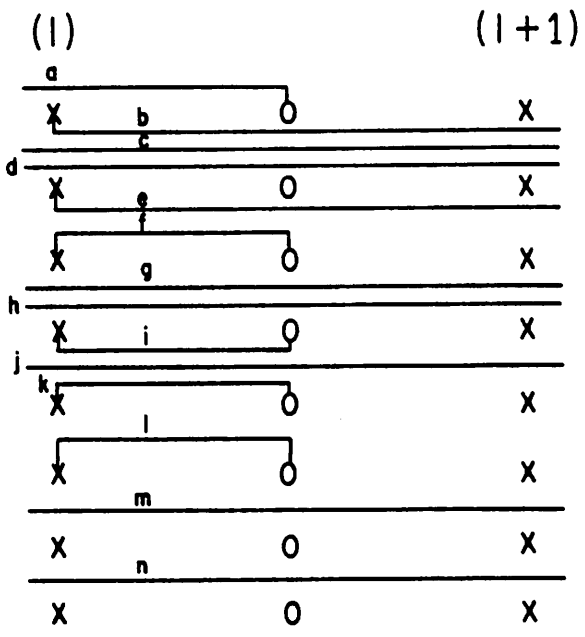

column β
column δ
row α
row γ
x o .... o x
⊕ x
o x

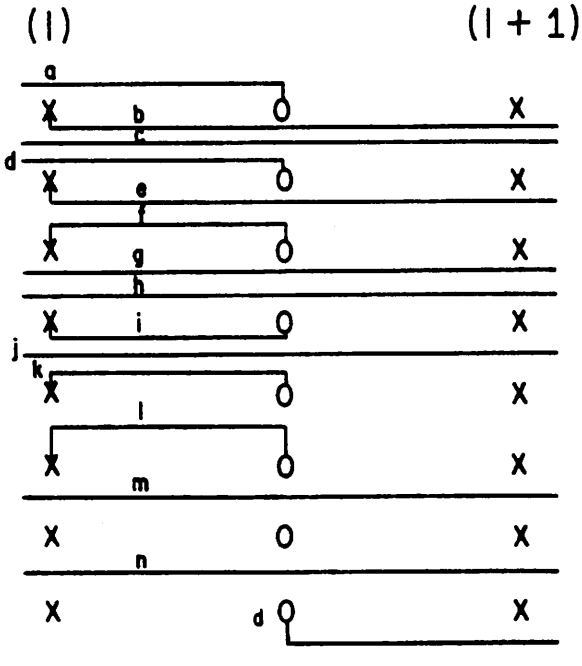

(l)
(l+1)
a
X b
c
d
X e
g X f
h
X i
j
X k
X l
m
X
n
X
0
X
f X
k X
a X
i X
l X

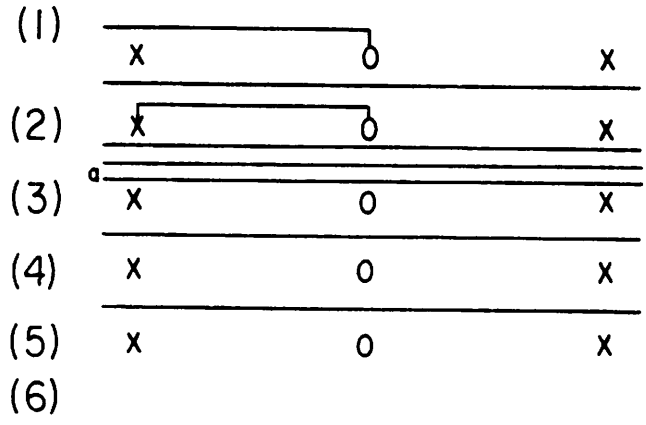

(l)
(l + 1)
a
X b
c
d
X e
f
g
h
i
j
k
l
m
X
n
X
0
X
f X
k X
a X
i X
l X

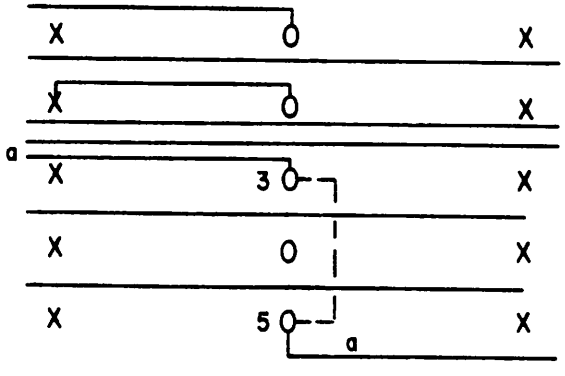

X
X
X
X
X
X
X
X
O a
O f
O i
O k
O l
O
O
O
X
X f
X k
X a
X i
X l
X
X

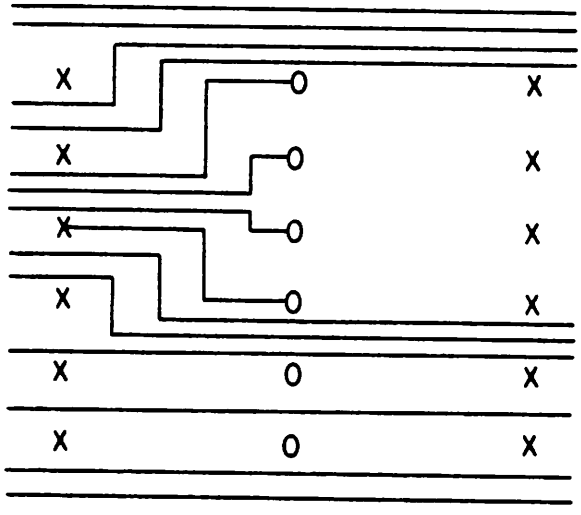

(1)
(a)
(2)
(b)
(3)
(c)
(4)
1
2
3
x
5
4
2
1
5
3

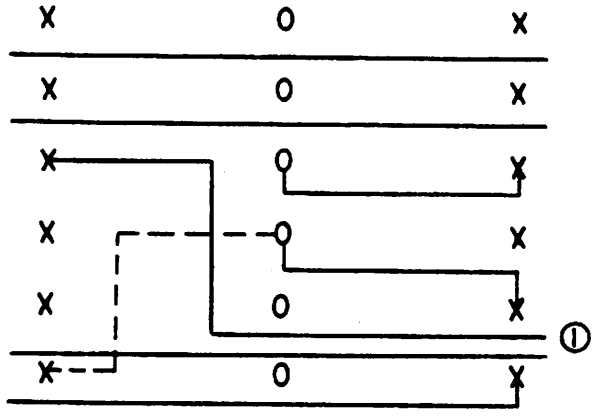

(l)
(l+1)
a
b
c
d
e
f
g
h
i
j
k
l
m
n
X
O


(l)
(l + 1)
a
b
c
d
e
f
g
h
i
j
k
l
m
n
d
X
O

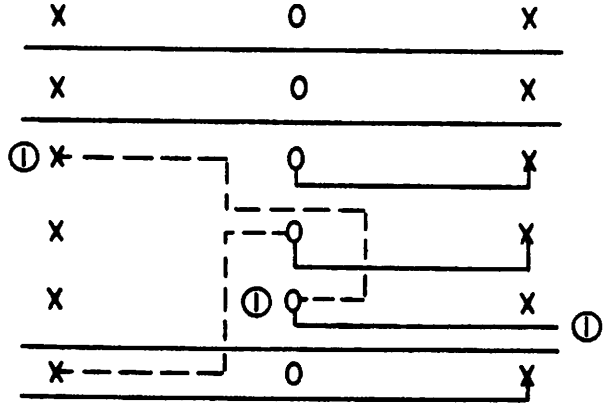

(1)
X
0
X
(2)
X
0
X
a
(3)
X
0
X
(4)
X
0
X
(5)
X
0
X
(6)

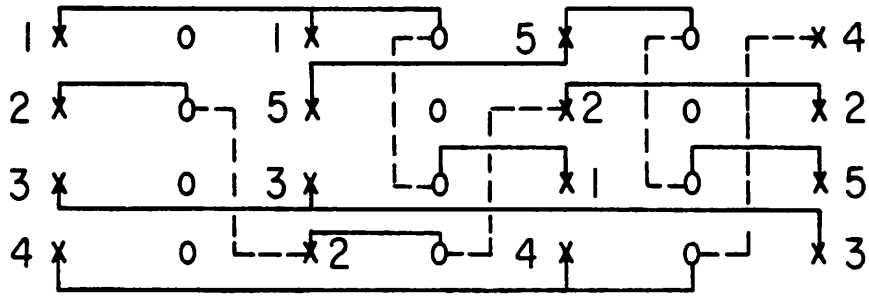

X
0
X
X
0
X
a
X
3
0
X
X
0
X
X
5
0
a
X

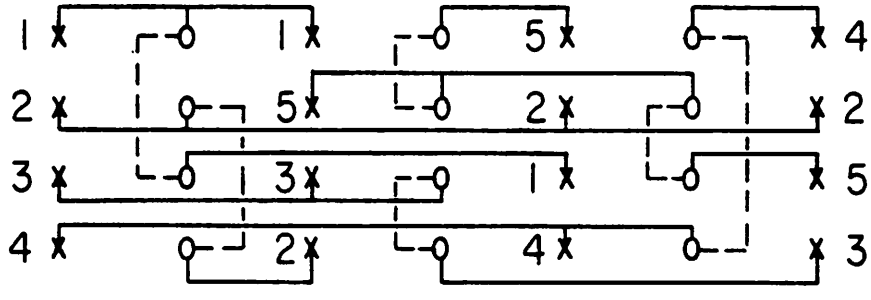

X
O
X
X
O
X
X
O
X
X
O
X
X
O
X
X
O
X

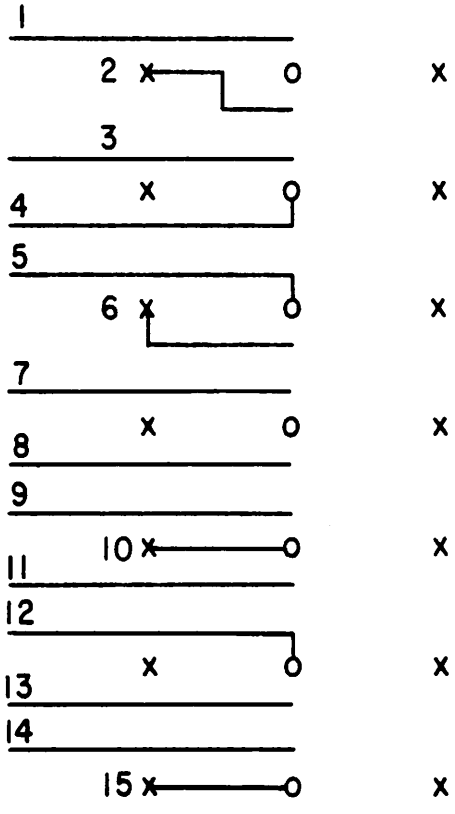


(a)




1
2
3
4
5


1
2
3
4
5

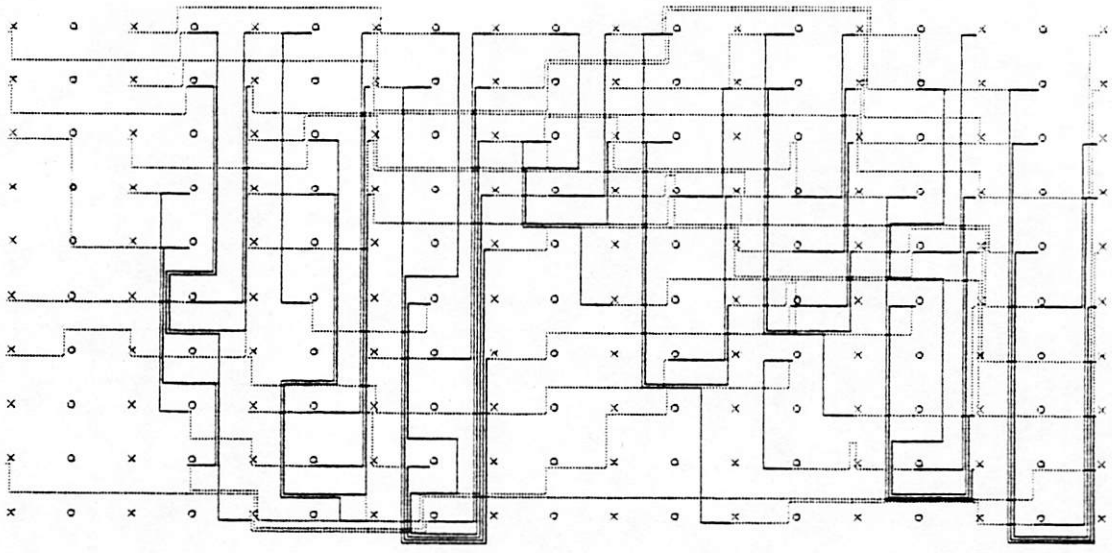


W1:

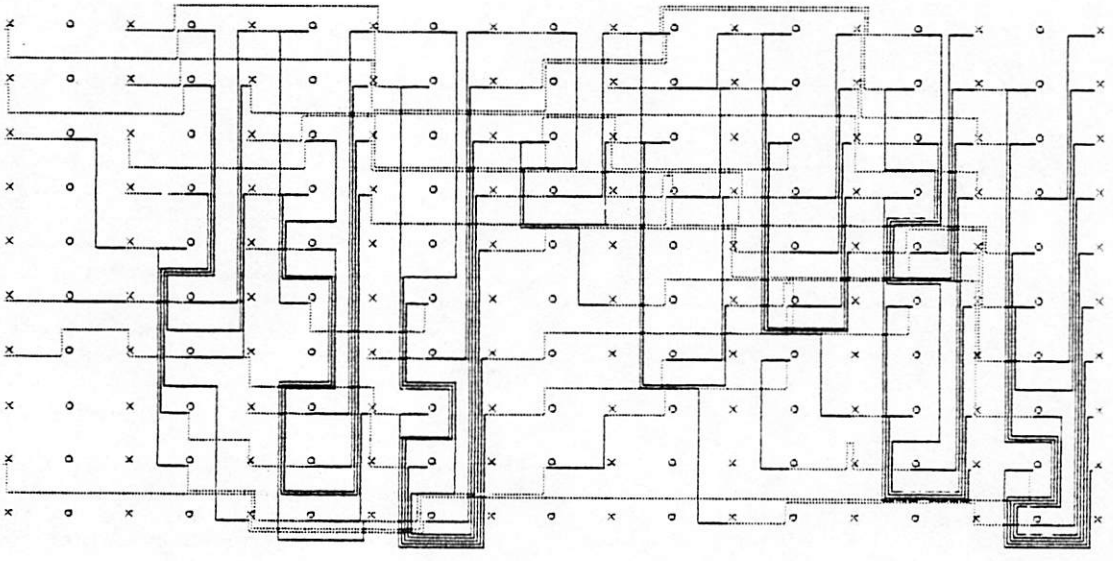
[1, -2, 3, 0, 4, 5, -6, 7, 0, 8,
9, -10, 11, 12, 0, 13, 14, -15]

W2:

[1, 0, 2, 3, -4, -5, 6, 7, 0, 8,
9, -10, 11, -12, 13, 14, -15]



(a)



(b)