

SU-SEL-75-049

SU-326P.39-6

A NEW PHILOSOPHY FOR WIRE ROUTING

by

B. Ramakrishna Rau

November 1975

Technical Report No. 101

The work described herein was supported by the U. S. Energy Research and Development Agency (formally AEC) under Contract AT(04-3)326P.A.39..

**DIGITAL SYSTEMS LABORATORY**

**STANFORD ELECTRONICS LABORATORIES**

**STANFORD UNIVERSITY . STANFORD, CALIFORNIA**



A NEW PHILOSOPHY FOR  
WIRE ROUTING

by

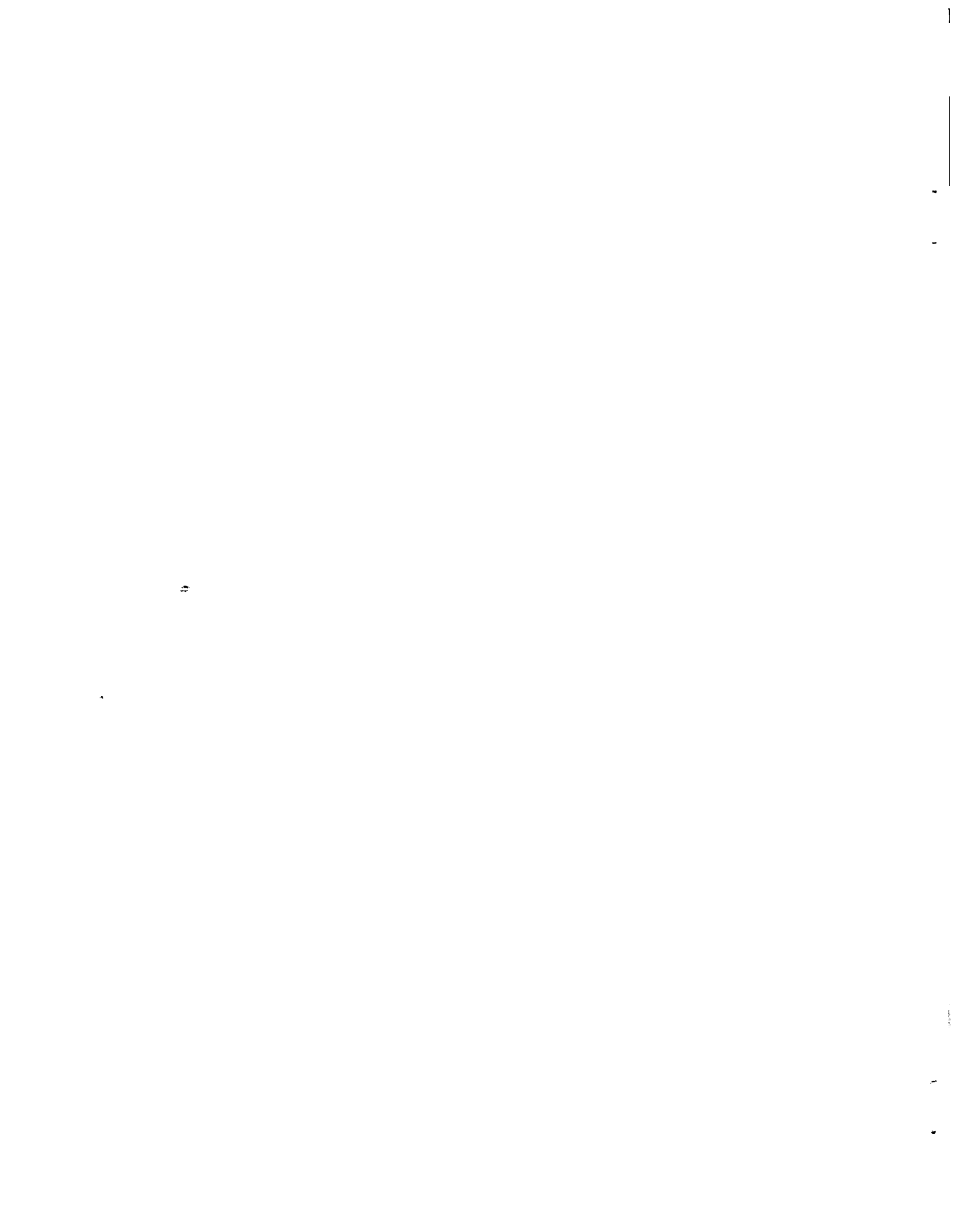
E. Ramakrishna Rau

November 1975

Technical Report No. 101

DIGITAL SYSTEMS LABORATORY  
Stanford Electronics Laboratories  
Stanford University  
Stanford, California

The work described herein was supported by the U.S. Energy  
Research and Development Agency (formally AEC) under  
contract AT(04-3)326P.A.39.



Digital Systems Laboratory  
Stanford Electronics Laboratories

Technical Report No, 101

November 1975

A NEW PHILOSOPHY FOR  
WIRE ROUTING

by

B. Ramakrishna Rau

ABSTRACT

A number of interconnection algorithms exist and have been used quite successfully. However, most of them, though differing in detail, appear to subscribe to the same underlying philosophy which has developed from that for single layer boards. Arguments are advanced which question the validity of this philosophy in the environment of multilayer board technology.

A new philosophy is developed in this report, which, it is hoped, will be more suited for use with multilayer boards. Based on this philosophy, an interconnection algorithm is then developed in a step by step fashion.



## 1. INTRODUCTION

Substantial effort has gone into the development of interconnection algorithms, resulting in a vast improvement in their performance. In most cases, however, this effort has been directed at refining one or other of the disjoint steps constituting the interconnection algorithm, and despite the fact that the environment has changed considerably, the underlying philosophy has remained relatively static. The most significant component of the change in the environment was caused by the introduction of multilayer boards. We shall address ourselves to the formulation of a philosophy suited to multilayered boards. In Section 2 we discuss current algorithms and the philosophy underlying them. We shall also point out some of their shortcomings. In Section 3 we shall outline the new philosophy and algorithm that are developed in this report. Based on this, in Section 4, we shall examine the means for incorporating this philosophy into a working algorithm.

In the context of this report, we shall consider the interconnection algorithm to be that part of the Design

Automation system which, starting with a net list, produces the wire layout for the layered board. In particular, placement, pin and gate assignment shall not be considered in any detail. However, we shall indicate how some of these functions may be absorbed into the algorithm which is to be developed here.

## 2. A REVIEW OF EXISTING ALGORITHMS

Most existing interconnection techniques are designed to execute in four relatively disjoint steps. They normally accept as input the net list, i.e. the list of the sets of pins that have to be connected to one another, **The** four steps are:-

### 2.1. WIRE LIST DETERMINATION

This step converts the net list to a wire list. This entails specifying the list of wires which will interconnect the pins belonging to a net. In general these pins may be connected by some type of tree or chain, An attempt is made to minimize the total length of the tree or chain, the rationale being that this will facilitate the subsequent steps, particularly the layout step. During the placement



phase, a positive correlation exists between the total interconnection length and the routability observed during the subsequent interconnection phase. Hence, the total length is a meaningful criterion during placement. However, it is not clear that the total wire length should be the prime consideration during the wire list determination step. An attempt at reducing the total length of a net to a minimum results in each net being interconnected independently of the other nets around it. One could, conceivably, end up increasing the number of intersections and, hence, degrading routability by placing excessive emphasis upon the total length. Stevens has observed [STEV72] that localized congestion might have more effect upon the routability of a board than has the total wire length. It is advisable, therefore, to concentrate more on equalizing the wire density over the board when determining the wire list.

## 2.2. LAYERING

If multiple layers are available for routing, an attempt is made to place on separate layers those wires that interfere most with one another. The goal here is to minimize the total amount of interference as defined by some

measure. The most common measure of interference between two wires is to check for a Euclidean intersection between them. Thus the measure of interference between two wires assumes the value 1 if both are on the same layer and if their Euclidean routes intersect, and 0 otherwise. This measure can be criticized on the grounds that since wires are normally laid out in a rectilinear fashion, the Euclidean measure is unduly severe because the Euclidean routes may intersect even though the rectilinear ones do not.

Accordingly, Abel [ABEL72A] and Rubin [RUBI73] have proposed measures of interference based upon rectilinear routes. The weight assigned to the interference between two wires is a function of the manner in which the Minimum Distance Rectangles (MDR) of the two wires overlap. As before, the measure of interference assumes this value only when both wires are on the same layer.

In either case, the aim of the layering step is to minimize the total interference between all wires taken two at a time by assigning them to layers appropriately\*. This could be stated as a problem in linear programming.

However, the cost of solving it this way goes up exponentially with the number of wires [RUBI73]. Instead, heuristic methods have been proposed to accomplish this task [ABEL72A],[RUBI73],[ROZE64] and [AKER72].

The problem with this approach to layering is that it is based upon insufficient and misleading information. Clearly, the layering has to depend upon the final layout of the wires. However, the initial layout of the wires, whether Euclidean or rectilinear, very often bears little resemblance to the final layout. This is particularly so with the longer wires. Furthermore, it is not sufficient to consider the interference between wires taken only two at a time. Wires which initially did not interfere might well do so when they are eventually routed. This is caused by non-intersecting wires which, nevertheless, interfere because they both pass through the same congested area. Routing one of them through that area might require that the other be displaced. This is a form of interference that is overlooked at the layering step with the result that non-intersecting sets of wires get assigned to the same layer in violation of capacity constraints,

### 2.3. ORDERING

This step is considered to be the critical one in most algorithm. The reason for this lies in the nature of the last step, the layout step. Typically the layout algorithm is topologically inflexible and is unable to foresee the consequences of the decisions that it makes, i.e., it lacks lookahead capability. This being the case, it is deemed essential that wires be laid out in the "optimum order". This means that the first wire laid out should be the one which has the lowest potential for interference with the remaining wires. This necessitates a measure for the "degree of interference" between wires. Measures for interference have been proposed by Akers [AKER72] and Abel [ABEL72A]. However, both these measures assume that two wires that do not intersect, or whose MDR's do not overlap, do not interfere. This assumption is generally not valid, since the route allocated to a wire, during the layout step, might end up passing through the MDR of a wire with which it originally had no conflict. Furthermore, wires that intersect have varying degrees of interference. For instance, if the point of intersection is close to either end of the wire, the interference is less than if the intersection were at the center. In addition, the

congestion in the vicinity of the intersecting pair is an indication of the effect on other wires of eliminating this intersection. A measure that took all these factors into account might be of some value, However, such a measure would not be amenable to measurement, In view of the inadequacy of the measures of interference, it is not surprising to find that the specific ordering used does not impact performance much. Abel [ABEL72B] claim very little difference in the performance of ordering schemes, some of which are diametrically opposed in their philosophies, indicating a lack of correlation between the measures of interference used and the actual potential for interference that might exist, Even in retrospect, having laid out a board in the best possible way, it is difficult to specify a rigid order in which the wires should be routed to produce the same layout since it often happens that two wires must each make way for the other in different areas of the board,. Routing either wire first would result in the other one being blocked, This would all seem to question the effectiveness of an a priori ordering when the routing is neither iterative nor has lookahead,

#### 2.4. LAYOUT

This final step is concerned with the actual layout of the interconnection paths on the card. In most cases this is carried out one wire at a time. The order in which the wires are laid out is specified by the ordering step. The most frequently used wire routing algorithm is Lee's algorithm. It is topologically inflexible and possesses no lookahead capability. For reasons already mentioned, the probability of one wire hindering the layout of a subsequent one is high. Hitchcock [HITC69] has proposed the Cellular Routing algorithm which is topologically flexible. Hashimoto and Stevens' [HASH72] Channel Assignment algorithm has the same virtue but requires the use of vias which detracts from its generality. Rubin [RUBI74] has described an iterative version of Lee's algorithm which, however, is not topological in nature. All these algorithms reduce the dependence of the final layout on the ordering and are, therefore, improvements on Lee's algorithm. The first two do not possess any lookahead. Rubin's iterative algorithm provides lookahead from the second iteration on, since, after the first iteration, most wires are on the board and can be used as an indication of the paths to avoid,

However, since the first iteration begins with an empty board, the convergence to the final layout is much slower than if a lookahead capability had been consciously included.

The fundamental problem with the current approach is that it has evolved from that for single layer boards. For interconnection on a single layer the approach, having obtained the wire list, is to route the wires, which, in turn requires an ordering step because of the non-iterative nature of most wire routing algorithms. When interconnecting on multilayer boards the approach has been to reduce it to several disjoint single layer problems. Hence the introduction of the layering step before the ordering step,

In the next section we shall attempt to develop from scratch a philosophy for multilayer boards,

### 3. DEVELOPMENT OF THE NEW PHILOSOPHY

The layering step depends heavily upon the ability to measure the degree of interference between wires. The best way to do so is to actually try and eliminate or at least to minimize the intersections between the wires. The tenacious intersections which refuse to be eliminated indicate strong interference between the pairs of wires causing them. At this point, the layout of the wires would, hopefully, be closer to their eventual layout. The layering step would be able to operate on the basis of more reliable information than it would have with the Euclidean or MDR representation of the wires. Of equal importance is the fact that the layout could be checked to ensure that the capacity of the board was not exceeded in any area. This would avoid the problems arising out of layering the wires in a manner such that more wires were assigned to a layer than that layer had capacity for.

By attempting to route the wires before layering we would, in effect, be considering the interference between wires taking into account all the factors mentioned earlier,



such as the density distribution of wires, the distance from the point of intersection to the end of a wire, etc. It is hoped that this approach will result in a layering that yields a higher completion rate.

It is true, too, that just as the layering is dependent on the final layout, so is the layout dependent upon the layering. Specifically, when laying out the wires, one need not worry about "intersecting" wires that will not end up on the same layer. However, in avoiding such intersections we merely err on the safe side and, perhaps, cause some circuitous interconnections. Putting the layout step first results in, at worst, some waste of effort, but layering first can result in unroutable layers..

Fundamental, then, to our new algorithm is the philosophy which requires us to route the wires before layering them. This is the major point of departure from previous approaches. In addition, we shall try and avoid the limitations that we have observed in some of the existing algorithms. They are summarized below:

1. The lack of a lookahead capability,
2. Topological inflexibility, i.e., the inability to

push wires aside to make way f'or another wire.

3. The inability to re-route wires.
4. Fragmentation of the interconnection problem into a number of disjoint steps resulting in local optimization.
5. The use of questionable performance measures for each of these disjoint steps,

Many of the algorithms described in the literature have addressed one or more of these points. However, we are not aware of any system that has attacked all of them. Beginning with the basic algorithm prescribed by our new philosophy, we shall, **step** by **step**, refine our algorithm until we have considered all these points,

#### ALGORITHM 1 .

1. Obtain the wire list from the net list. Attempt to provide a fairly uniform distribution over the board of the wire density.
2. Route the wires **on** a single layer so as to obtain the minimum number of intersections, without exceeding the **capacity** of any region.
3. Layer the wires.

While forming the wire list we bear in mind the correlation between the existence of congested areas and difficulty in routing the wires. Accordingly, rather than concentrating on minimizing the total net length, we shall try and equalize the wire density over the board. The routing is accomplished by one of the wire routing algorithms mentioned in the literature, We shall not, at present specify which one and shall return to this issue in the next section. We proceed with the routing as though we had just one layer, but we account for the availability of  $L$  layers by multiplying the capacity of every area on the layer by  $L$ . Thus, if we were using Lee's algorithm, we would permit upto  $L$  wires to pass through every square on the grid. Once again, we do not at present specify the layering algorithm to be used and shall postpone it to the next section,

The algorithm in its present form is not satisfactory. It suffers from all the limitations that we have listed, Our first refinement will be to provide a little

intelligence to the algorithm in the form of a lookahead capability. Step 2 of the previous algorithm is replaced by the following:

ALGORITHM 2

1. Obtain the wire list from the net list, Attempt to provide a fairly uniform distribution over the board of the wire density.,
- 2.1. Lay out all the wires using canonical paths.
- 2.2. Order the wires.
- 2.3. Route the wires in the above order so as to minimize the number of intersections and ensuring that no capacity is exceeded,
3. Layer the wires,

If the layout is to be performed by a rectilinear routing algorithm, the canonical path is two sides of the MDR. When we introduce topological routing, the canonical path will be the Euclidean straight line joining the two pins. Now, during the routing of any wire, whether it be the first or the last one, we have all the other wires laid

out on the board. The algorithm can attempt to avoid paths that are potentially troublesome. If step 2.1 is thought of as a first pass and 2.3 as a second pass at routing, we see that we have provided a certain amount of iteration, Step 2.2 requires heuristics which can only be validated empirically. If past experience with ordering is to be believed we should not expect the ordering to impact performance substantially as long as we do not resort to any obviously foolish policy. For the time being we shall assume that the wires are ordered by some rule which tends to place long wires, with many intersections, at the top of the list. Consequently, short wires get re-routed last and tend to maintain their original short paths while the long wires tend to avoid the congested areas. This approach is superior to the previous one in that, first, the long wires try to avoid the short wires. If unable to do so, the short wires later will try to avoid the long ones. This reduces to some extent the problems arising from one-wire-at-a-time routing with a rigid ordering.

The algorithm still suffers from the fact that a definite ordering exists with the attendant worry that this was, perhaps, not the best ordering for this particular

board. We attempt to reduce the dependence upon the ordering by using a topological routing algorithm such as Hitchcock's Cellular routing algorithm which assigns a topologically distinct path but does not tie the wire down to a specific physical path. This permits the router to "move" wires aside and makes the layout step less sensitive to the order specified, Step 2.3 is altered to read as follows:

#### ALGORITHM 3

1. Obtain the wire list from the net list. Attempt to provide a fairly uniform distribution over the board of the wire density.
  - 2.1. Lay out all the wires using canonical paths,
  - 2.2. Order the wires.
  - 2.3. Topologically route the wires in the above order so as to minimize intersections without exceeding any capacity.
3. Layer the wires,

Our last refinement to the routing step is to make it iterative, thereby removing almost all dependence on the

order specified insofar as the final layout is concerned, The speed of convergence to this final layout will definitely depend on this ordering, and we shall, therefore, retain this step. The procedure used is to repeatedly re-route the wires in the order specified until no further improvement is observed. The question that next arises is whether the order should remain static or should change with each iteration, This is a heuristic that is best decided empirically. The algorithm reflecting these changes is:

#### ALGORITHM 4

1. Obtain the wire list from the net list, Attempt to provide a fairly uniform distribution over the board of the wire density,
  - 2.1. Layout the wires in their straight Euclidean paths.
  - 2.2 Order the wires.
  - 2.3 Topologically re-route the wires in the order specified,
  - 2.4. If any improvement was observed then repeat 2.2 or 2.3,
3. Layer the wires,

The above algorithm developed out of an attempt to approach the layout problem in the way a human -- or rather, this human -- might have. The natural line of attack seemed to be to picture the wires as rubber bands stretched out between the corresponding pairs of pins. Intersections would then be eliminated or minimized by stretching one of the intersecting rubber bands over the pin of the other. The choice of which wire of the pair to move would depend upon which pin was closer to the point of intersection. Use would be made at each point in time of the current layout of all the other wires, and the process would naturally be iterative. Accordingly, the layout step developed here has been affectionately dubbed the Rubber Band Algorithm.

We shall now discuss the layering step. Presumably, we have eliminated all intersections that could be removed without exceeding the capacity of any area of the board (multiplied as it is by a factor of  $L$ ). Our only recourse now is to eliminate the remaining intersections by assigning intersecting wires to different layers. Care would have to be taken to not assign wires to a layer such that the capacity of any region is exceeded. We would have to modify



the layering algorithm of our choice to check for capacity violations and flap: these as interference in addition to the more obvious interference in the form of intersections.

The layering step can be improved by making it iterative. By doing so we reduce its dependence on the order in which the wires were originally considered for assignment. We now have the following algorithm:

#### ALGORITHM 5

1. Obtain the wire list from the net list, Attempt to provide a fairly uniform distribution over the board of the wire density.
  - 2.1. Layout the wires in their straight Euclidean paths.
  - 2.2 Order the wires.
  - 2.3 Topologically re-route the wires in the order specified.
  - 2.4.. If any improvement was observed then repeat 2.2 or 2.3,
3. Layer the wires iteratively.

Needless to say, we shall still have a few intersecting wires on each layer at the end of all this. This might be the result of having tried to route the wires before layering. Consequently, we permitted certain intersections which we should not have in an attempt to eliminate intersections which we need **not** have since those wires subsequently went on to different layers. Our sins have caught up with us! We shall try to rectify matters by re-routing wires within the assigned layer. A side benefit of this step is that wires which are unnecessarily circuitous (another consequence of routing before layering) can assume more direct paths:

..

#### ALGORITHM 6

1. Obtain the wire list from the net list, Attempt to provide a fairly uniform distribution over the board of the wire density.
- 2.1% Layout the wires in their straight Euclidean paths.
- 2.2 Order the wires,
- 2.3 Topologically re-route the wires in the order specified,

2.4. If any improvement was observed then repeat 2.2 or 2.3.

3. Layer the wires.

4.1. Order the wires on each layer.

4.2 Use the Rubber Band Algorithm on each layer.

The last refinement to the layering step is to apply iterative layering with simultaneous re-routing to wires which give us difficulty. Such wires would be re-routed on all layers and assigned to that on which it has fewest intersections. The final version of our algorithm is:

#### ALGORITHM 7

1. Obtain the wire list from the net list. Attempt to provide a fairly uniform distribution over the board of the wire density,
  - 2.1. Layout the wires in their straight Euclidean paths.
  - 2.2 Order the wires.
  - 2.3 Topologically re-route the wires in the order specified.
  - 2.4. If any improvement was observed then repeat 2.2

or **2.3.**

3. Layer the wires.

4.1. Order the wires on each layer.

<sup>3</sup> 4.2 Use the Rubber Band Algorithm on each layer.

5. Repeat the layering step (3) with the modification that each candidate for reassignment is tentatively assigned to each layer and given its minimum intersection path. The wire is finally assigned to the layer on which it has fewest intersections.

By introducing steps 4 and 5 we have attempted to reduce the effect of the fragmentation caused by having distinct routing and layering steps (2 and 3). The temptation to avoid fragmentation entirely by omitting steps 2, 3 and 4 is strong. We would then have one big iterative loop which would perform layering and routing at the same time. This would take a tremendous amount of time to converge. The algorithm in its present form permits faster convergence, and the existence of step 5 removes the effects of the fragmentation in steps 2 through 4.

#### 4. IMPLEMENTATION

In this section we shall discuss the specific algorithms to be employed to implement the Rubber Band Algorithm and the layering.

The Rubber Rand Algorithm requires a wire routing algorithm which will meet the following requirements:

1. It routes the wires topologically, i.e., it fixes the pins between which the wire passes, but does not fix the precise points through it passes.
2. It checks whether the capacity of any area will be exceeded by routing the wire through it,
3. It should be capable of generating all, or most, of the possible paths available for routing the wire,

Hitchcock's Cellular algorithm [HITC69] is the best choice since it meets all three requirements, Hightower's algorithm [HIGH69] could be made to meet the first condition by fairly minor modifications. Gut the enhancements necessary to allow it to make capacity checks would make it

look very much like the Cellular algorithm. Mah and Steinberg's Topological Class routing algorithm [MAH72] falls short in that only certain types of interconnection paths are permitted. The Channel Assignment algorithm [HASH72] developed by Hasimoto and Stevens requires that each layer have paths of only one orientation, either horizontal or vertical. Such a routing strategy requires a large number of vias. We would prefer to develop an algorithm which attempts an interconnection with no vias at all if possible. If the Channel Assignment algorithm were generalized to permit both horizontal as well as vertical paths on the same layer, then the capacity check would be complicated sufficiently to be similar once again to Hitchcock's scheme.

We shall, therefore, consider two wire routing algorithms -- the Cellular routing one and a line routing algorithm which may be thought of as an extension of either Hightower's algorithm or of Hasimoto and Stevens' with the necessary capacity checks added.

The layering problem has been given an excellent treatment by Rubin [RUBI73]. His conclusion is that the

best layering algorithm is the Fast Assignment one developed by him. This algorithm is iterative in nature and will suit us perfectly, The Fast Assignment algorithm is outlined below:

1. Assign all wires to layer 1 and establish a table showing the number of intersections each wire would have on each layer, Order the wires on each layer in a linear list in the order of decreasing number of intersections.
2. Choose layer 1 as the current layer,
3. Choose the first wire in the list of the current layer as the current wire.
4. Find from the table the minimum number,  $M$ , of intersections that it has on any layer.
5. If  $M$  is less than the current number of intersections reassign the current wire to the first layer on which it has  $M$  intersections. Attach it to the bottom of the corresponding layer list. Update the intersection table.
6. If  $M$  equals the current number of intersections then

set the current wire aside for consideration later.

7. Choose the next wire from the current layer list as the current wire if all the wires have not yet been considered and repeat from 4.
8. Choose the first wire that was set aside as the current wire..
9. If on any layer it has fewer or an equal number of intersections as on the current layer then assign it to the first such layer. Attach it to the bottom of the corresponding layer list, Update the intersection table.
10. Repeat from 8 until all wires set aside have been tried in the order in which they were set aside.
11. If on any of the last L layers a reassignment was made then choose the next layer as the current layer. Repeat from 3.
12. stop\*

The algorithm as listed here differs from Rubin's in the introduction of layer lists. This ensures that a wire which has just been assigned to a layer will be the last candidate for reassignment.



## 5. A SAMPLE PROBLEM

Since this algorithm has not yet been implemented, a simple problem, which could be solved manually, has been used. Fig.1 is the sample board to be interconnected on two layers,

Fig.2 shows the result of first layering, using the Fast Assignment algorithm, and then routing. The intersections were inevitable since the layering step, on the basis of the straight line intersections, assigned wires without regard to the capacity. Figs.3 through 5 are the results of Algorithms 1 through 3, the layering being effected by the Fast Assignment algorithm in all cases. A steady improvement may be noted. The relative costs of the algorithms may be deduced from the statistics presented for the number of **lines** routed (more than the number of wires if iteration is employed), the number of wires inspected during layering and the number of wires that are reassigned to the other layer. Fig.6 shows the result of Algorithm 6.

Fig.7 is the result of reconfiguring the two nets consisting of wires C,J and H,E respectively to minimize the congestion through the central part of the board,

## 6. ADDITIONAL FEATURES

So far the newly developed algorithm has attempted to interconnect pins without the use of vias. Whereas the operation of this algorithm does not hinge upon the availability of vias, their presence can help in routing the last few wires for which the algorithm has been unsuccessful in finding a path without intersections. If the vias are fixed, an attempt is made to find a path passing through vias such that no section of the path (the portion between two vias) intersects wires on every layer. If floating vias are available the problem is simplified. The best route found so far for the wire is divided up into sections such that no section intersects wires on every layer. The vias are introduced at the endpoints of each section.

Another possibility is that an **attempt** be made to modify the net list for the net which includes the problem wire. An alternate path could be found to connect the net which is left disconnected by the deletion of the problem wire.

Use could also be made of the existence of equivalent pins to facilitate the layout step. If two wires going to equivalent pins intersect, the intersection can be eliminated by exchanging the pins. This is quite simple if a topological routing algorithm is being used. This applies both to pins on a package as well as to external pins. Thus we see that the external pin assignment should be frozen only after the layout step.

## 7. CONCLUSION

Arguments have been advanced for questioning the traditional approach to the interconnection problem. A new philosophy has been advanced which is more suited to multilayered boards. An interconnection algorithm has been developed based on this philosophy.

The algorithm is perfectly general in that it applies equally well to single layer boards as to multilayer boards. When used with single layer boards, it would reduce to the Rubber Band algorithm, and would possess the desirable features of lookahead, iteration and topological flexibility.

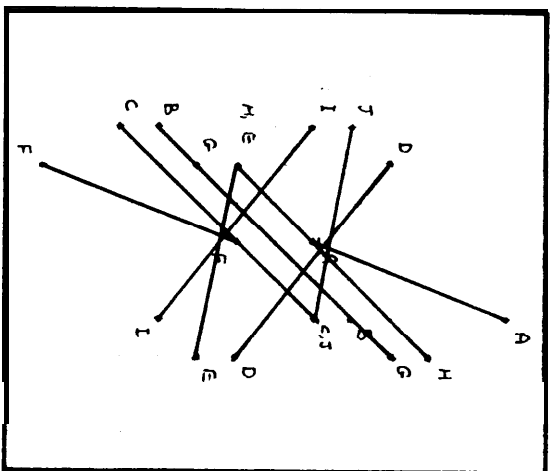
Work is currently in progress to implement and evaluate this algorithm.

4

## BIBLIOGRAPHY

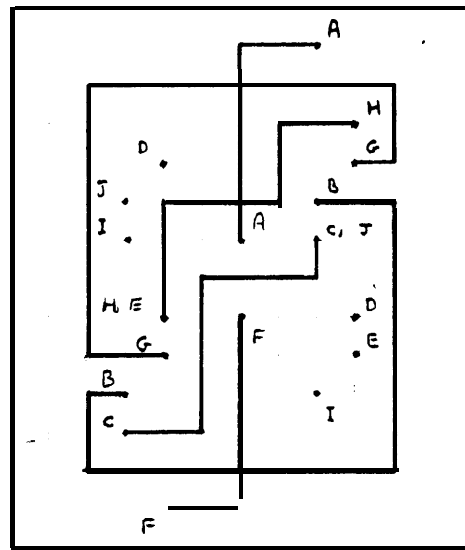
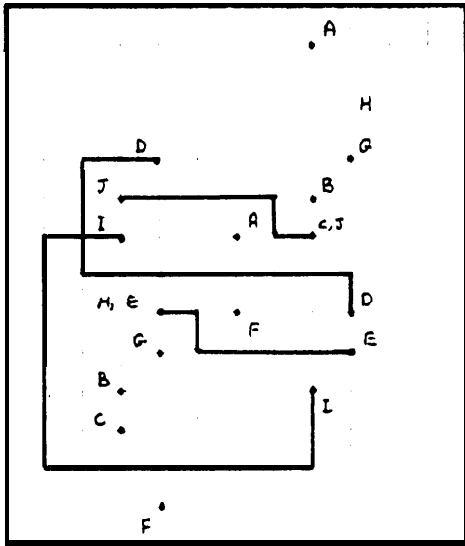
- ABEL72A Abel,L., "On the Automated Layout of Multi-Layer Planar Wiring and a Related Graph Coloring Problem". Coordinated Science Laboratory Report No. R-546, Univ. of Ill., Jan. 1972.
- ABEL72B Abel,L., "On the Ordering of Connections for Automatic Wire Routing". IEEE TC, Nov. 1972.
- AKER72 Akers,S., "Routing." Design Automation of Digital System, Vol. 1, Prentice-Hall, 1972.
- FISK67 Fisk,C.J., et al, "Topographic Simulation as an Aid to Printed Circuit Board Design." 4th Design Automation Workshop, 1967.
- HASH72 Hashimoto,A.,Stevens,J., "Wire Routing by Optimizing Channel Assignment Within Large Apertures," 9th Design Automation Workshop, 1972.
- HIGH69 Hightower,D.W., "A Solution to Line Routing Problem on the Continuous Plane." 6th Design Automation Workshop, 1969.
- HIGH73 Hightower,D.W., "Interconnection Techniques - A Tutorial? 10th Design Automation Workshop, 1973.
- HITC69 Hitchcock,R.B., "Cellular Wiring and the Cellular Modelling Technique." 6th Design Automation Workshop, 1969.
- LASS69 Lass,S., "Automated Printed Circuit Routing with a Stepping Aperture," CACM, May 1969.
- LEE6 1 Lee,C., "An Algorithm for Path Connections and its Applications." IRE Transactions on Electronic Computers, Sep. 1961.
- MAH72 Mah,L., and L.Steinberg, "Topological Class Routing for Printed Circuit Boards? 9th Design Automation Workshop, 1972

- RUBI73 Rubin, F., "Assigning Wires to Layers of a Printed Circuit Board." 10th Design Automation Workshop, 1973.
- RUBI74 Rubin, F., "An Iterative Technique for Printed Wire Routing." 11th Design Automation Workshop, 1974.
- STE72 Stevens, J.E., "Fast Heuristic Techniques for Placing and Wiring Printed Circuit Boards." Dept. of Computer Science, Univ. of Illinois at Urbana-Champaign, No. UIUCDCS-R-72-558, Oct. 1972.



10 wires  
to be assigned on 2 layers

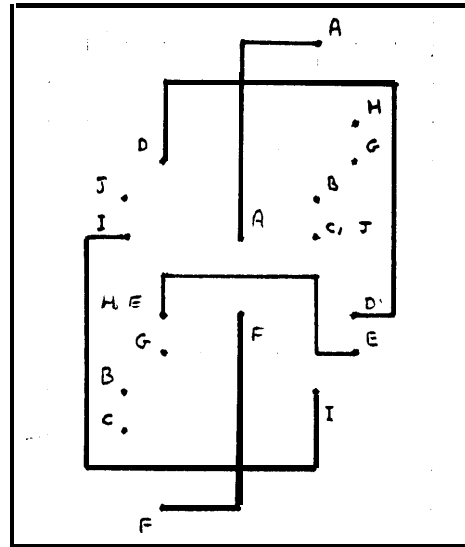
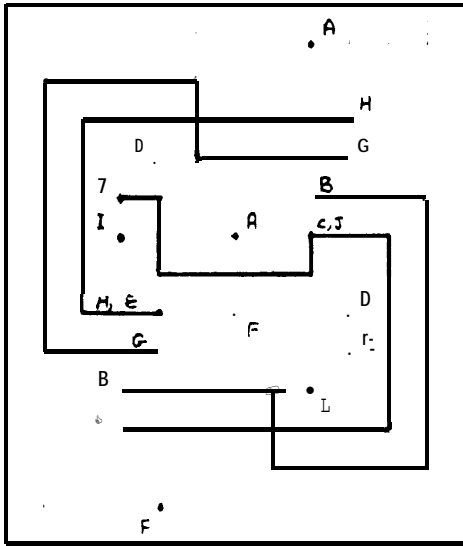
FIG 1



10 wires routed  
 27 wires inspected  
 6 wires reassigned

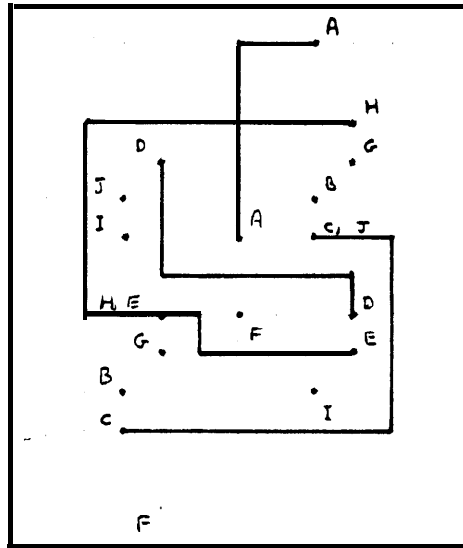
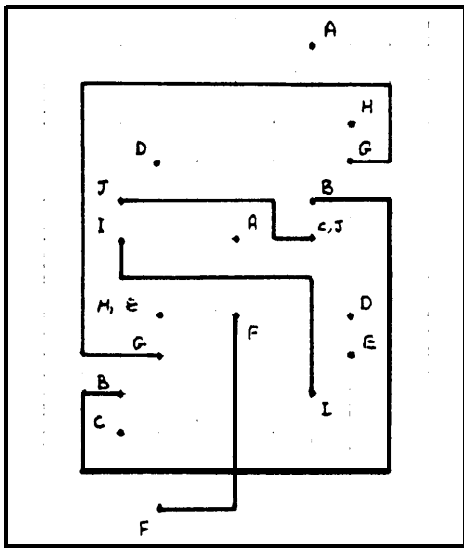
FIG. 2





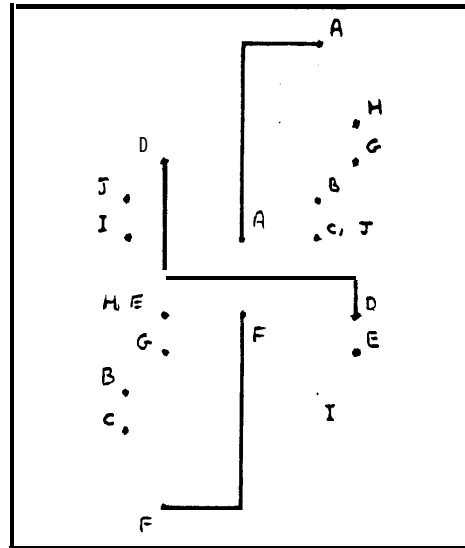
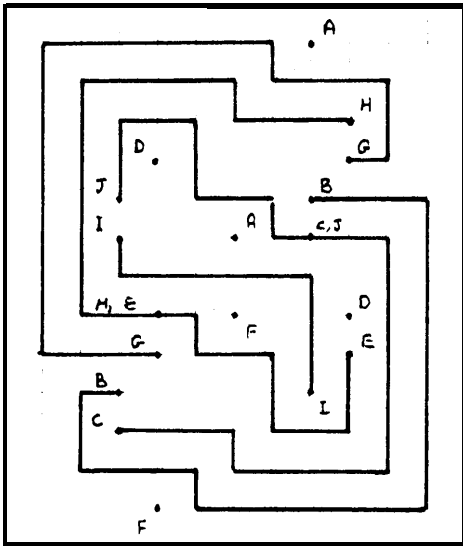
10 wires routed  
 24 wires inspected  
 6 wires reassigned

FIG. 3



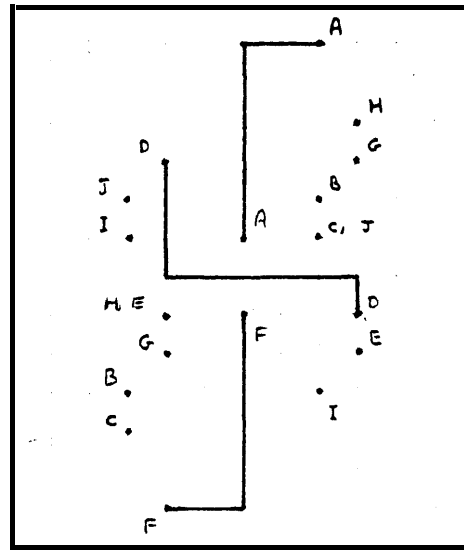
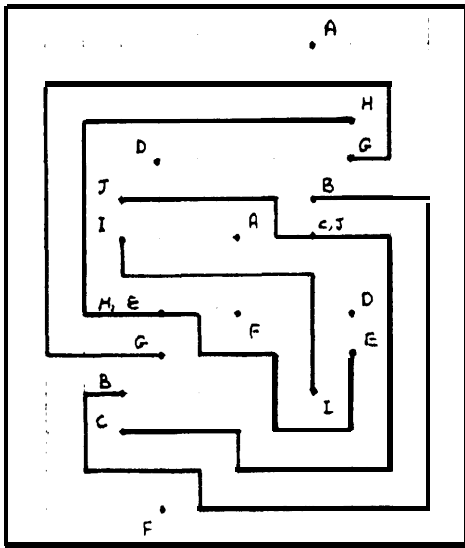
20 wires routed  
 37 wires inspected  
 11 wires reassigned

FIG. 4



20 wires routed  
 20 wires inspected  
 3 wires reassigned

FIG. 5



30 wires routed  
 20 wires inspected  
 3 wires reassigned

FIG. 6



