AN ALGORITHM FOR AUTOMATED PLACEMENT AND ROUTING USING BARYCENTRIC EMBEDDING

by

CYNTHIA LEW

ELECTRICAL ENGINEERING

Submitted in Partial Fulfillment of the Requirements of the University Undergraduate Fellows Program

1981-1982

Approved by:

Dr. Douglas M. Green

May 1982

TABLE OF CONTENTS

ABSTRACT

This study presents an automated integrated circuit and printed circuit routing algorithm based on barycentric embedding. This algorithm deals with circuits which are represented as graphs. In order to utilize this barycentric embedding technique, it is necessary for the graphs to be planar and 3-connected. The algorithm involves a modification of the Hopcroft and Tarjan planarity test and their algorithm for dividing a graph into 3-connected components. The development of this algorithm is a preliminary step to the implementation of a computer program which will perform the automated routing function.

This thesis was prepared in the format of the IEEE Transactions on Computers.

INTRODUCTION

Automated layout of integrated circuits can be subdivided into three tasks: 1) partitioning, 2) placement, and 3) routing. Partitioning involves grouping logical components into cells; these cells are determined such that there are a minimum number of interconnections between the cells. Placement attempts to minimize the distance between highly connected the cells. Routing determines the specific path used to connect two cells. It is desired to minimize the number of crossovers and the total length of the paths.

Current routing algorithms typically use a Lee router or a modification of it to determine the interconnection paths. The Lee router guarantees that the shortest path between two points will be found, if a path exists. The problems with this router are the large memory space that is required, and the fact that the method used is not efficient and thus requires excessive time to complete a routing problem. Many times not all the interconnections can be made, thus requiring some connections to be made by hand.

The algorithm discussed here is one that will carry out the entire layout procedure with a series of computer programs. This procedure is based on an algorithm by W. T. Tutte [1] for barycentric embedding. Barycentric embedding will be used to carry out the placement and routing problems. In order to use this procedure, the circuit of interest must be represented as a graph or a group of subgraphs. The graph(s) must possess two necessary characteristics before the barycentric embedding algorithm can be used [1]. First, the graph must be planar. A planar graph is a graph that can be embedded on the surface of a sphere without any crossed edges [2]. Secondly, the graph must be 3-connected. The connectedness of a graph is determined by the number of edges that must be removed in order to split one graph into two disjoint graphs [3]. For 3-connectedness, it is necessary to remove three edges in order to create two disjoint graphs. It is the purpose of this paper to describe an algorithm which can be used to find planar and 3-connected graphs.

BACKGROUND

Many algorithms have been written that will test a graph to determine whether it is 3-connected or not [4]-[8]. The first recursive algorithm for testing for planarity was proposed by Auslander and Parter [4]. This method had an error and could have looped indefinitely. An algorithm by A. J. Goldstein corrected the flaw by using iteration instead of recursion [5]. This algorithm uses depth-first search to determine the order of calculations and thereby achieve efficiency. Through the use of depth-first search an initial cycle is found and deleted. The remaining disjoint paths will be tested to determine if their addition to the initial cycle will result in a planar graph. Recently, depth-first search has been used to construct efficient algorithms to solve problems in graph theory. J. Hopcroft and R. Tarjan have utilized the depth-first search in many of their algorithms. One algorithm will determine whether a given graph is planar [6] and a second will divide a graph into triconnected components [9]. The planarity test uses a procedure that requires a rearrangement of the paths if the graph is initially determined to be nonplanar.

It has been shown by Rubin that rearrangement of the paths is unnecessary if the paths are considered in the proper order [7]. Since rearrangement is not necessary, this algorithm has been run in half the time necessary to implement the Hopcroft and Tarjan algorithm. First, any cycle is chosen and imbedded. Next the paths that begin and end on the initial cycle are found and imbedded. If the paths can be imbedded either inside or outside the mesh, these paths are imbedded inside. If the path cannot be imbedded either inside or outside the mesh, the graph is called nonplanar.

Another algorithm for determining planarity, by John Bruno, is based on the structural characterization of planar graphs [8]. In this algorithm the original graph is decomposed into smaller pieces that are triply connected. This concept is discussed in a paper by Lane [10]. A graph is planar only if its triply connected components are planar. The first step in this algorithm is a reduction of the original graph. If only two edges share a common vertex then the two edges are in series. Series edges can be compressed into a single edge by disregarding the vertex that was originally shared. When two edges share the same endpoints and have no other edges branching off either parallel edge, one of the edges can be removed. These two reductions are successively carried out until the graph cannot be reduced further. Once the reduction is complete the graph can be checked for planarity.

Hopcroft and Tarjan have devised an algorithm for dividing a graph into 3-connected components [9]. This is a necessary algorithm in that it can provide graphs that meet one of the necessary requirements for barycentric embedding.

BARYCENTRIC EMBEDDING

Barycentric embedding is the procedure that will be used to determine the exact placement and the routing pattern that will be used. This procedure is based on W. T. Tutte's paper "How to draw a graph" [1]. Barycentric embedding can only be applied to graphs that are both planar and 3-connected. Assuming the graphs of interest fit both requirements, the details of barycentric embedding is discussed below. A program for barycentric embedding has been written by Robert Dawes [3]; this program will be available when the project has advanced to that stage.

The input to this program will be a planar, 3-connected graph. Each of the vertices will be assigned a relative weight determined by the size of the component represented by the vertex and the adjacent space that must be allocated to each component. Vertices that represent large components will be given relatively small weights. Likewise, small components, those that require only a small amount of space in the embedding will be assigned a heavier weight. These weights will determine the exact location that the vertices will occupy.

The bounding polygon of the graph will be extracted and embedded on the peripheral of a circle. Each of the vertices will be equally spaced on the circle. The remaining vertices will be embedded in the circle at a location that will be determined by the adjacent vertices. The interior vertices will be placed at the center of gravity between its two adjacent vertices. The center of gravity is determined according to the weights that were initially assigned to each vertex. Using this procedure, all the planar, 3-connected graphs can be embedded.

OBTAINING PLANAR, 3-CONNECTED GRAPHS

The application for automated routing lies in printed and integrated electrical circuits. It is known that the graphical representation of any worthwhile circuit is nonplanar. Knowing this, it is desirable to obtain the largest planar subset of a given nonplanar graph that can be determined. Then the Hopcroft and Tarjan algorithm for dividing a graph into 3-connected components can be used. This will result in graphs that will fit the necessary requirements of planarity and 3-connectedness. The problem therefore reduces to the determination of one of the largest planar subset of a nonplanar graph.

The algorithms now presently available must be modified since they can only determine whether a graph is planar or not. The proposed algorithm will involve modifications of the available planarity tests. The information necessary for describing a graph will be stored in the form of an adjacency matrix. An adjacency matrix is an nxn matrix, where n is the number of vertices v, in which a_{ij} is one if v_i is adjacent to v_j and a_{ij} is zero otherwise. This structure has been found to be an efficient way to access the data necessary for working with graphs [6],[9].

The first step to be taken in obtaining a planar graph will involve a reduction of the original data set. Any vertices that have only one edge leading from them will be removed along with the edge. Figure 1a and 1b show the original graph, before reduction, and the graph after reduction of single vertex, edge pairs. Series and parallel reductions will also be carried out. These reductions will be the same type used by Bruno [8]. An example of these reductions can also be seen in Figure 1. It is of great importance that any type of reductions used do not affect the planarity of the graph. The three types of reduction used here do adhere to this requirement. The sole purpose of these reductions is to decrease the data set necessary to describe the graph. It is desired that such a reduction of the working data set will reduce the running time of the algorithm.

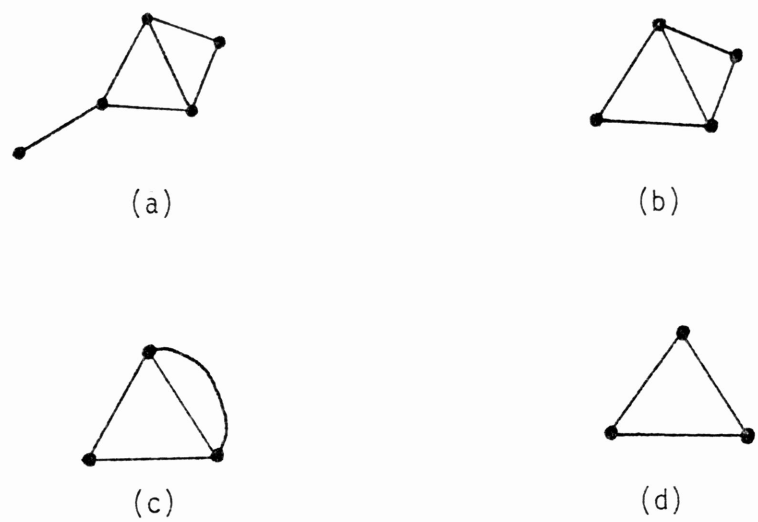

Figure 1.(a)A simple graph before any type of reduction; (b)Graph after single edge reduction;(c)Graph after series reduction;(d)Graph after parallel reduction.

The Hopcroft and Tarjan algorithm for testing for planarity [6] provides the basis for the proposed algorithm and will be discussed here. A lemma states that a graph is planar if $E \leq 3V - 3$ where E is the number of edges and V is the number of vertices. It should be noted that if a graph passes this test it does not guarantee that the graph is planar. If the graph in question does not pass this test, the algorithm will terminate and define the graph as nonplanar.

Next a depth-first search will be performed. The depth-first search will convert the graph into a palm tree and number the vertices. At this point the Auslander and Parter algorithm is used to find an initial cycle and then delete it. The deleted cycle will not be represented as a closed graph but as a palm tree. The remaining disconnected edges will be tested for planarity with the initial cycle. If it is determined that a path plus the original palm tree can be embedded in a plane, the path will be added to the left side of the palm tree. The remaining paths will be tested and added to the palm tree if they are suitable. If the paths cannot be placed on the left side of the palm tree and still maintain planarity, it will be placed on the right side. If the path can be placed on neither the left nor the right side of the palm tree, the existing paths will be rearranged in an attempt to be able to place the new path on the palm tree. If, after an attempt at rearrangement, the path still cannot be added to the palm tree and maintain planarity, the graph will be classified as nonplanar and the algorithm will terminate.

The desired algorithm will, following the reduction, order the vertices using a depth-first search. An initial cycle will be found and each of the disjoint paths will be tested for planarity as in the Hopcroft and Tarjan algorithm. The modification of the Hopcroft and Tarjan algorithm is that when the original algorithm terminates with a nonplanar graph the new algorithm will not terminate but continue the process. The path being tested at the time of termination will be deleted from the data set. In order to continue the processing the graph in question, the routine will have to be reinitialized or have the ability to simply continue with the next path that was to be chosen. At this time the exact procedure for continuing the test is not known.

At this point the resulting palm tree will be a large planar subset of the original nonplanar graph. It is now desired to break this large planar graph into a set of planar, 3-connected graphs. As previously mentioned, Hopcroft and Tarjan also devised an algorithm for dividing a graph into 3-connected components [9]. This algorithm will not be discussed here. Once this algorithm is applied to the planar graph, the resulting graphs will be planar and 3-connected. It will now be possible to use barycentric embedding to determine the actual placement and routing for each individual subgraph.

LEE ROUTER

Initially, the graph was reduced in an attempt to decrease the actual working set. Those edges and vertices that were removed must now be taken into account. The paths that caused the palm tree to be classified as nonplanar were eliminated. Also, it may be necessary to delete some edges and vertices when dividing the graph into 3-connected components. All of these edges and vertices must be routed using an alternative methods. The method chosen here will be a Lee router. Lee routers of it are currently being used to perform the entire routing problem. Lee routers guarantee that the shortest path between two points will be found, if a path exists. Lee routers are not efficient and thus not desirable for large scale problems. The number of vertices involved here is expected to be small enough that efficiency will not be of major concern. When these remaining connections are made the entire graph will have been routed.

CONCLUSIONS

The algorithm presented here offers new alternatives to the routing problem. It incorporates many aspects of graph theory techniques. Due to time limitations the project was not implemented in computer code. Once the code has been written, the true feasibility of this approach can be tested. Whether the program will be able to handle a wide range of graphs and how efficient this procedure is will be determined when the code is written. It is expected that the computer code will be completed by August 1983.

REFERENCES

[1] W. T. Tutte, "How to draw a graph," Proc. London Math. Soc. (3)13(1963)743-68.

[2] R. J. Wilson, Introduction to Graph Theory, second Edition, Academic Press, 1979.

[3] R. L. Dawes, Personal Correspondence.

[4] L. Auslander and S. V. Parter, "On imbedding graphs in the plane," J. Math. and Mech. 10,3(May 1961)517-523.

[5] A. J. Goldstein, "An efficient and constructive algorithm for testing whether a graph can be embedded in a plane," Unpublished.

[6] J. Hopcroft and R. Tarjan, "Efficient planarity testing," J. of the Association for Computing Machinery, Vol. 21, No. 4 , October 1974, 549-568.

[7] F. Rubin, "An improved algorithm for testing the planarity of a graph," Trans. on Computers, Vol. C-24, No. 2, February 1975, 113-121.

[8] J. Bruno, K. Steiglitz, and L. Weinberg, "A new planarity test based on 3-connectivity," IEEE Trans. on Circuit Theory, Vol. CT-17, No. 2, May 1970, 197-206.

[9] J. Hopcroft and R. Tarjan, "Dividing a graph into triconnected components," Siam J. Computing, Vol. 2, No. 3, September 1973, 135-158.

[10] S. Mac Lane, "A structural characterization of planar combinatorial graphs," Duke Math J., Vol. 3, September

1937, 460-472.