

Computer-Aided Design of MIC Layouts with Postprocessor for Photoplotter

K.R. Ramesh Kumar

Defence Research & Development Laboratory, Hyderabad-500 258

ABSTRACT

A PC-based computer-aided design (CAD) package for designing microwave integrated circuit (MIC) printed circuit board (PCB) layout, called microwave artwork postprocessor for photoplotter (MAPP), is developed. The small size of MIC PCBs makes them suitable for defence applications. MAPP has a postprocessor which converts the MIC layout data to a format acceptable as input to a Gerber photoplotter. Photomask of the PCB is obtained from the photoplotter. This paper presents the methodology for the CAD of MIC layouts and also discusses the algorithm for filling of a general polygon with curved edges, and which lies in the real domain. This algorithm is used in the postprocessor for a photoplotter.

1. INTRODUCTION

Microwave integrated circuit (MIC) printed boards (MIC PCBs) are particularly desirable for defence applications because of their small size and also that most MICs are planar in structure. Amplifiers, filters, multiplexers, couplers and mixers are some of the components which can be realized using MIC PCBs. MICs also find applications in radars, radiometry and seeker for missile guidance. Once an MIC is designed, it must be translated to an artwork from which photomasks must be generated. The process for generating a photomask aided by a computer is shown in Fig. 1. In computer-aided microwave artwork design, the engineer directly draws his MIC PCB layout on a graphics terminal using a graphics editor. He then uses a postprocessor which converts the MIC layout data to a format acceptable as input to a photoplotter. The photoplotter directly exposes a photofilm to generate the photomask. The $N:1$ artwork cutting and $1:N$ photoreduction stages involved in manual photomask generation are eliminated in computer-aided photomask generation. This makes computer-aided

photomask generation five to ten times faster and also more accurate as compared to manual photomask generation.

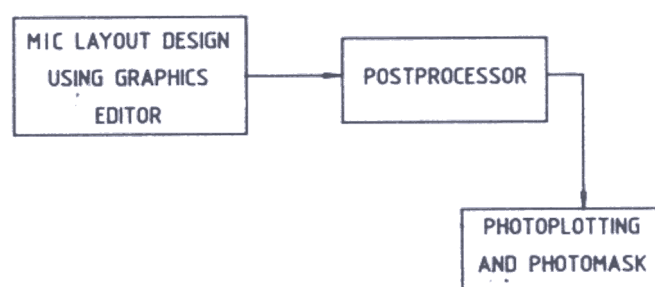


Figure 1 Flow diagram for generating photomask aided by a computer.

A computer software MAPP (microwave artwork postprocessor for photoplotter) has been developed on IBM PC using PASCAL language for computer aided design (CAD) of MIC PCB layouts. This paper discusses the method used by MAPP software for CAD of MIC PCB layout design. Polygon filling algorithm used in the postprocessor for a photoplotter is also presented.

2. METHODOLOGY USED FOR CAD OF MIC LAYOUTS

MAPP software can be divided into three sections, namely, (i) user interface (ii) graphics editor and (iii) output section¹. Each section is closely integrated with the other using a shared database to form the complete package.

2.1 User Interface

MAPP is totally menu-driven and user-friendly. The screen of MAPP when activated is shown in Fig. 2. The screen is divided into three areas. All options can be chosen from menu area. Command area responds to

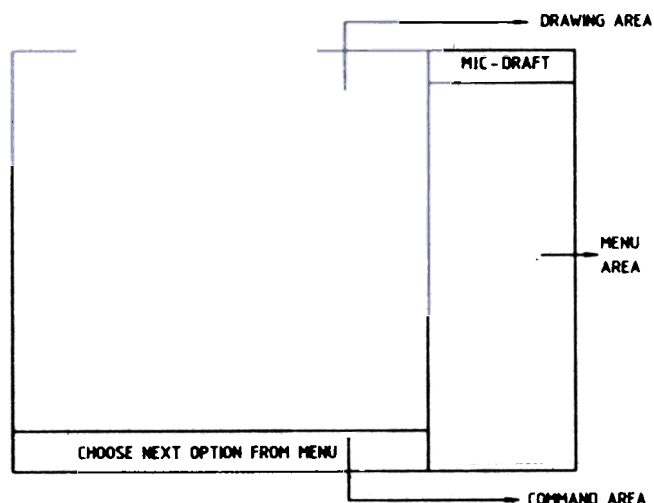


Figure 2 Screen of MAPP.

options chosen from menu area and prompts the user for inputs. Design rule violations, such as invalid geometric parameters are conveyed to the user in this area. Drawing area is the region where the graphics can be viewed and updated.

2.2 Graphics Editor

MIC layout design requires a set of graphics primitives to draw filled or unfilled shapes. MAPP provides a graphics editor which allows the user to graphically edit his MIC artwork by converting user coordinates into screen coordinates. Geometric primitives are represented by their parametric equation defined for some finite extent. Explicit functions can be multivalued and so are not suitable for programming languages. The single valued nature of parametric functions makes it desirable for programming languages.

2.3 Outputs

MAPP can give outputs on both printer and photoplotter. Printer outputs can be obtained by converting bitmapped screen graphics to printer format file. MIC layout data are converted to Gerber format data for photoplotter outputs. This is achieved through a postprocessor.

3. FILLING OF A GENERAL POLYGON WITH CURVED EDGES AND WHICH LIES IN THE REAL DOMAIN

Polygon filling has vast applications in CAD/CAM and CAE. One application is 'shading' in raster graphics, to illustrate a solid. Scan conversion algorithm is frequently used for area filling. In raster graphics, vertices of the polygon lie in integer domain. Another application for polygon filling is a postprocessor that generates NC code for an NC machine or a photoplotter. For such an application, vertices of the polygon lie in the real domain. The filling algorithm for such a case is described here.

3.1 Filling Algorithm

Filling technique to fill a polygon with curved edges involves discretisation of each curved edge into small line segments which join successive points on the curve. The algorithm to achieve polygon fill is as follows:

Once every curved edge is discretised, scan conversion algorithm is used for area filling. The algorithm proceeds in three steps:

1. Generate a linked list of all vertices of the polygon.
2. From the vertex list, generate an edge list for all edges. This is the most important step of polygon filling as it involves treatment of vertices as being an extreme or a non-extreme.
3. Repeat the following steps for scanline = Y_{\min} to Y_{\max} in steps of delta (Y_{\min} and Y_{\max} are minimum and maximum Y-coordinate values of the polygon, delta is the step size):
 - (a) For the current scanline value compute the X-coordinate intersection values of the scanline with all edges;
 - (b) Sort edge intersections in increasing order; and
 - (c) Group edge intersections by pairs and join correct pairs by lines.

3.2 Vertex Considerations

In Fig. 3 every scanline has an even number of edge intersections. The number of intersections is two for every scanline other than the first and the last ones for

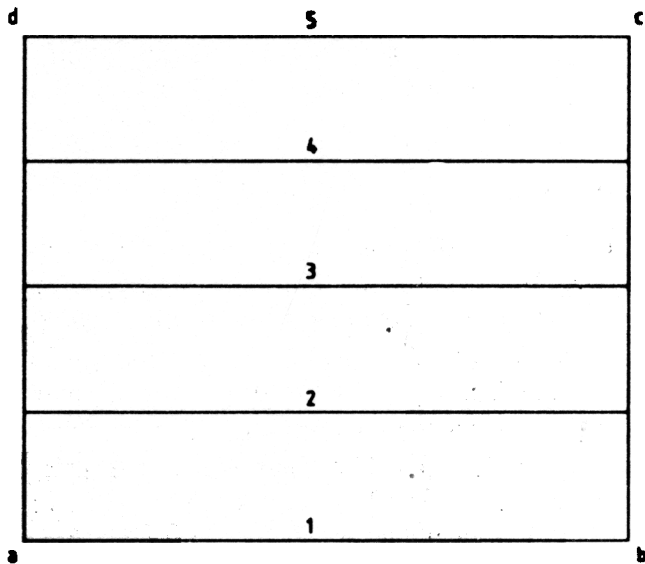


Figure 3. Filling of a simple convex polygon.

which it is four. Polygon in Fig. 4, however, does not yield an even number of edge intersections for every scanline. Scanline-2 yields three edge intersections. So does scanline-3. Step (3) of the fill algorithm fails in these cases.

It may be noted that both these scanlines pass through vertices of the polygon. But so does scanline-5.

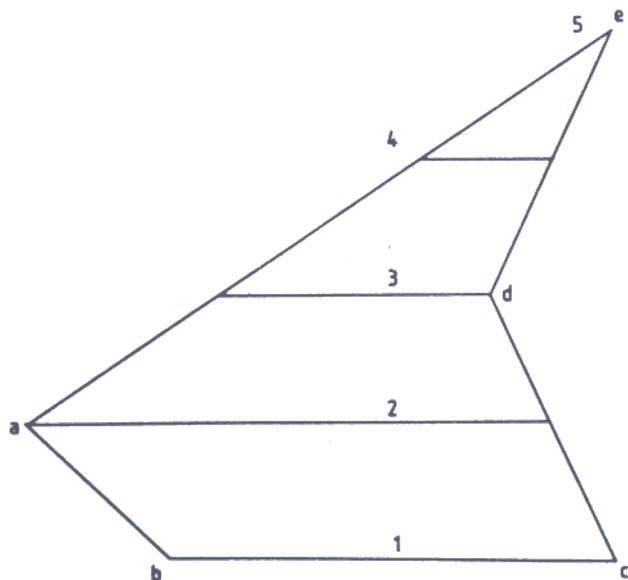


Figure 4. Filling of a concave polygon.

Vertex-e of the polygon is an extreme edge as it connects edges of increasing and decreasing slopes. Such vertices are counted twice. Vertex-a and vertex-d are non-extremes. They are counted only once (consequently resulting in even number of edge intersections). To ensure this, the lower end of the upper edge is slightly shortened by a small value.

3.3 Algorithm for Vertex Considerations

Whenever a non-extreme vertex is encountered, it must be counted only once. The algorithm to achieve this is as follows:

- (a) If $Y(K) > Y(K+1)$ (Fig. 5(a)) then
 If $Y(K+1) < Y(K+2)$ (Fig. 5(b)) then
 No additional processing of edge
 else shorten lower end of upper edge ($Y(K+1)$) (Fig. 5(c));
- (b) If $Y(K) < Y(K+1)$ (Fig. 5(d)) then
 If $Y(K) < Y(K-1)$ (Fig. 5(e)) then
 No additional processing of edge
 else shorten lower end of upper edge ($Y(K)$) (Fig. 5(f));

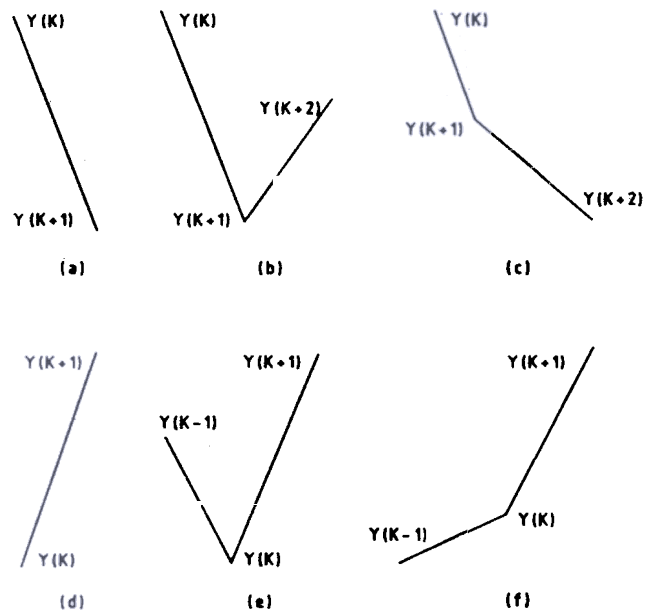


Figure 5. Vertex considerations for polygon filling.

The above algorithm removes horizontal edges (zero slope) from the edge list. These edges reappear when outline of the polygon is redrawn.

The general polygon filling algorithm can be modified to fill multiple polygons.

3.4 Data Representation and Memory Considerations

3.4.1 Data Representation

Dynamic data structure is used for data representation. Vertex is represented by a record shown in Fig. 6 called *vertex_type*. The first field 'V_info' is a coded integer entry and it contains information on vertex, like geometric entity type, whether starting or ending point and other related characteristics of entity.

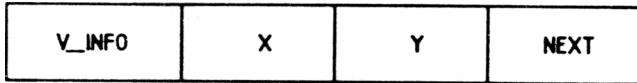


Figure 6. Data structure for vertex entry (*vertex_type*).

The second and third fields are real type, and are the 'X' and 'Y' coordinates of the vertex, respectively. 'Next' is a pointer and points to the next vertex. Arc, ellipse and exponential curve have additional attributes. In such a case, the same vertex type record is used to represent them but with a different name. This data structure runs parallel to the main vertex list (Fig. 7).

Once each curve of the polygon is discretised, the vertex list is modified to contain only the vertices of line. The same *vertex_type* record is used. 'V_info' field only indicates whether the vertex is start or end point of the line. Now the vertex list is simplified and is equivalent to that shown in Fig. 7, but to the left of the vertical dotted line.

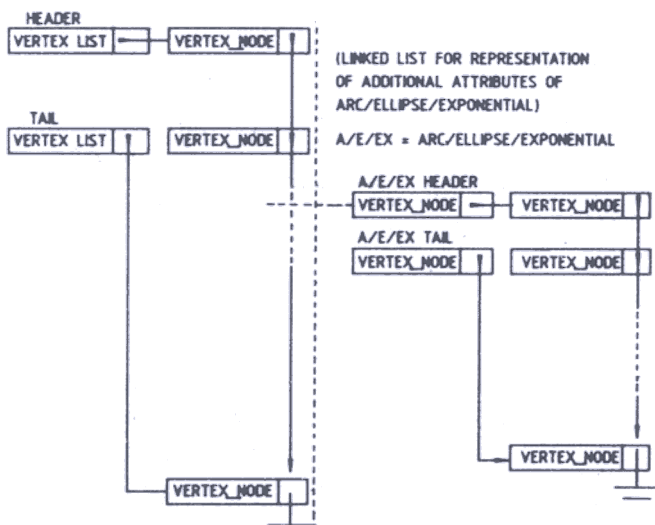


Figure 7. Representation of singly linked vertex list before discretisation of curved edges.

An edge list is generated from the vertex list and contains total information on an edge. X-intersections for a particular scanline can be obtained by running through each edge of edge list and testing for intersection. A sorted (increasing X) X-intersection list is then formed for a particular scanline. X-intersection values are paired properly to get area fill coordinates. By representing area fill coordinates in the Gerber format, postprocessed data file can be obtained.

3.4.2 Memory Considerations

It can be safely assumed that after memory is allocated to DOS, the main program, and other RAM resident programs, the remaining memory would be less than 400 kB. For the polygon filling algorithm, once the curved edges of polygons are discretised, the number of vertices becomes very large. In such a case there is every possibility of a heap overflow during program execution. To avoid such an error, the memory occupied by the lists must be deallocated as and when they are not required. Also overlays need to be used to circumvent this problem.

4. RESULTS

Several MIC layouts were designed successfully on MAPP. MIC layout of a Finline mixer designed on MAPP along with its photomask^{4,5} is shown in Fig. 8. The circuit layout is postprocessed using MAPP, based

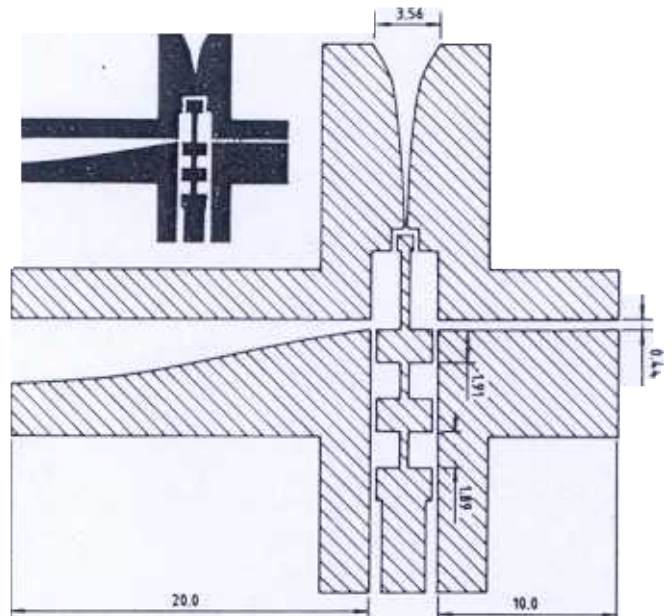


Figure 8. MIC layout and photomask of finline mixer.

on algorithm described in Sec.3. Photomask is generated on a Gerber(3235) photoplotter with a 4-mil pen. Dimensional accuracy of the photomask (determined by the photoplotter) is found to be between 2 to 15 micron.

5. SUMMARY

The method used for CAD of MIC PCB layouts is described. The polygon filling algorithm used in the postprocessor for a photoplotter is explained. Results are given for a circuit layout designed on MAPP.

ACKNOWLEDGEMENTS

The author is grateful to Dr R Balakrishnan, Director, DOVD and Shri AK Kaushik, Head, CAD/CAM Centre, for their guidance and encouragement provided during software development.

REFERENCES

- Newman, W.M. & Sproull, R.F. Principles of interactive computer graphics, Ed. 2. McGraw-Hill, New York, 1979.
- John, Rankin R. Computer graphics software construction using pascal language. Prentice-Hall Advances in Computer Science Series, Australia, 1989.
- John, Rankin R. A note on multi-polygon area filling. *Computer Graphics*, 1987, **11**(4), 445-47.
- Verghese, Thomas. Design and development of a crossbar mixer at Ka-band. I.I.T., Delhi. 1989. PhD Thesis.
- Ramesh Kumar, K.R. Computer-aided microwave artwork design with inbuilt postprocessor for photoplotter. DRDL, Hyderabad, 1992. DRDL-5231-0009-000.