Electrocomponent Science and Technology, 1980, Vol. 7, pp. 181–186
0305–3091/80/0701–0181$06.50/0

A NEW APPROACH TO THE TOPOLOGICAL DESIGN OF HYBRID CIRCUITS

WALTER ULBRICH

Lehrstuhl für Netzwektheorie und Schaltungstechnik, TU München, Arcisstraße, 21, D-8 München 2

and

RUDOLF VAN DER LEEDEN

D-8153 WEYARN

(*Received September 3, 1980*)

A computer-aided topological hybrid layout-design procedure is proposed, that yields the wanted principal routing in the form of a geometrical planarization graph. A so-called grid-embedding of a circuit graph into the Euklidean plane enables us to observe all except one of the various technological constraints. The real problem is reduced to finding a proper arrangement of "nets" and "flocks" in the plane in order to meet the omitted cross-capacity constraint. The solution is accomplished by a constructive and implicit enumeration procedure, which is used within an interactive man-machine design process.

1. INTRODUCTION

In this paper we deal with a problem which is of central importance in the design of hybrid microelectronic circuits. The development of a hybrid circuit layout is a complicated and skilful task, and only a few attempts have been made to use a computer in the essential parts of the layout-design process.[1,2,3] We restrict ourselves only to that part of the twofold layout-problem, called *topological design*, which ends up in a principal realizable routing of the conductors without regarding the size, the shapes and the positions of the elements. The second subsequent step would be the topographical design, that definitely yields the geometrical layout.

Solving the topological layout-problem, several technological and user-imposed constraints have to be taken into account. We assume only *one* layer for the realization of the circuit in thick film or thin film technology. Therefore, no crossings between different conductors and no over-lappings between components are allowed. The linear order of the terminals of line elements and the oriented cyclic order of the terminals of ring elements must be observed. Boundary terminals must be realizable at the boundary of the substrate in an eventually prescribed order. A very useful but constrained freedom is the possibility of crossing a gap between two adjacent terminals of a hybrid element by a limited number of conductor lines. According to Rose,[4] this number is called the cross-capacity of the gap, while the fact itself is referred to as the capacity-constraint.

The computer-aided solution of the topological hybrid layout-problem presented in this paper is accomplished by the following steps:[5,6,7]

1) MAPPING of the circuit into the *circuit graph S*.

2) GRID-EMBEDDING of S into the Euklidean plane R^2, generating a geometrical *planarization graph G*. The elements of G are arranged in R^2 by means of straight lines of an orthogonal grid. This helps in following the various constraints. Only the capacity constraint is not taken into account.

3) PLANARIZATION OF G, performed by using the freedoms of rearrangement within the frame of the grid. A graph G is successfully planarized and then is called a *feasible solution* of the topological layout-problem, if the capacity-constraint is also observed. This step is the main part of the procedure. We propose an algorithm for the constructive and implicit enumeration of all possible "net"-arrangements, while the "flock"-arrangement will be done by hand by means of man-machine interaction.

4) Optionally one can look for an improved result with respect to some criteria, e.g. the number of crossings used in an actual topological layout. This

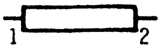
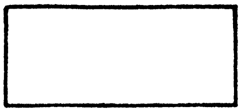
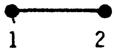
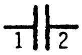
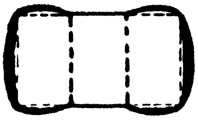
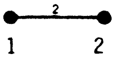
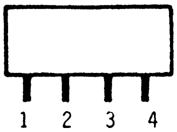
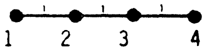
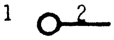
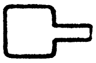
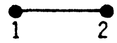
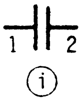
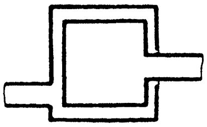
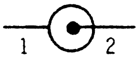
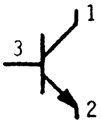
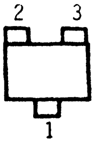
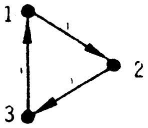
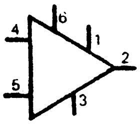
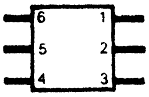
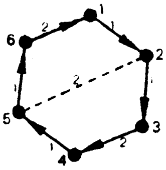
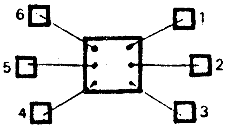
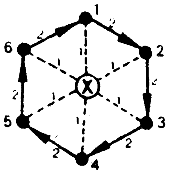
B	circuit element	component type	T	topological type	circuit graph (flock)
1	2	3	4	5	6
a			S	line-element	
b			H	line-element	
c			H	line-element	
d			S	line-element	
e			S	point-element	
f			H	ring-lid-element	
g			H	ring-lid-element	
h	see row g		H	ring-star-element	

FIGURE 1 Mapping of typical circuit elements into flocks of the circuit graph S, taking into account the actually related component types.

step of OPTIMIZATION is only possible, if a feasible solution has already been found.

The planarization algorithms are implemented on a HP 2100 computer with 32K words of memory in conjunction with the intelligent graphical terminal IMLAC PDS4. The following sections describe details of the different steps. A practical example and related results of the topological design is given in the last section.

2. CIRCUIT GRAPH *S*

Only topologically relevant features of a given circuit are modelled in the circuit graph *S*. We distinguish between line and ring elements depending on the component types actually used.[8] Figure 1 shows examples of either type. The general mapping rules are as follows:

– Each terminal is mapped into a unique *vertex* of *S*;

– Each terminal gap constrained by a cross-capacity equal or greater zero is modelled by an *edge*; all edges of a circuit element form a *flock*, which is the image of this element in *S*;

–Each connection tree is mapped into a hyperedge,[9] called *net*, which connects all vertices of the terminals joined by the tree.

Figures 1a to 1h show eight flocks of typical circuit elements and their component types. The cross-capacities are related to the edges. Oriented closed edge-trains are directed. The control of inner gaps is achieved by introducing inner edges (Figures 1g and 1h). The boundary of the substrate is imagined as a connection tree that connects all boundary terminals of the circuit. Accordingly, the boundary net links all 1-vertices of the boundary flocks (Figure 1d).

3. PLANARIZATION GRAPH *G*

Think of a grid in the Euklidean plane R^2 consisting of horizontal straight lines (*rows*) and vertical straight lines (*columns*). Rows and columns are identified by their y- and x-coordinates, respectively. The crossing of the row y and the column x defines the grid-point (x,y). The x-limited subplane between two columns is called a column-band. The *grid-embedding* of a circuit graph *S* generates a planarization graph *G* (also called a grid-model) and is performed in two steps. At first, nets are assigned to different adjacent rows and flocks are assigned to disjunctive adjacent column-bands. The boundary net (no. 1) is fixedly assigned to the row 1. The other assignments can be done arbitrarily. This important freedom will be used in the next section for the planarization of *G*. Here we choose the "natural" assignment, i.e. flock *i* is assigned to the column-band just after that of flock *i*-1 and net *j* is assigned to row *j*. The second step is the definite fixing of the flocks and nets. See Figure 2 an an example. Basically, vertices of *S* are represented in *G* by grid-points and edges by straight lines between two adjacent columns. Each column contains at most one vertex. Linear and cyclic orders are realized by according sequences of columns in the grid. A closed edge-train consists of a forward-running open edge-train and a closing two-pieces edge with a vertical section and a backward-running horizontal section. See for example the grid-embedding of flock 3 in Figure 2. A uniform clockwise orientation of all closed edge-trains can be achieved by choosing in each case the backward-*below* the forward-running part. All vertices connected by a net lie in a row and can be linked together simply by a limited straight line that defines the *living* section of the net within its row. A cross-point of a living net and an edge represents a crossing of a conductor line and a terminal gap. Each crossing reduces the cross-capacity of the participated edge.

The planarization graph *G* is called a *solution*, because all constraints except the capacity-constraint are taken into account. In general, this leads only to an *infeasible* solution. Feasibility is possible if for every edge in *G* its actual capacity is not less than zero.

4. PLANARIZATION

The above mentioned freedoms can be used for planarizing the geometrical graph *G*. The question is whether a net- and flock-arrangement exists that yields a feasible solution which strictly satisfies *all* constraints. At the moment, this problem has been solved by an inter-active procedure, in which the designer proposes a flock sequence while the search for an appropriate net-arrangement is performed automatically by a constructive and implicit enumeration algorithm.

Starting with a given flock-arrangement we look for a feasible net-arrangement. Let n be the number of nets in *G*, then $(n - 1)!$ possibilities of different arrangements have to be inspected for an exhaustive search. Boundary net 1 lies fixedly in row 1. We

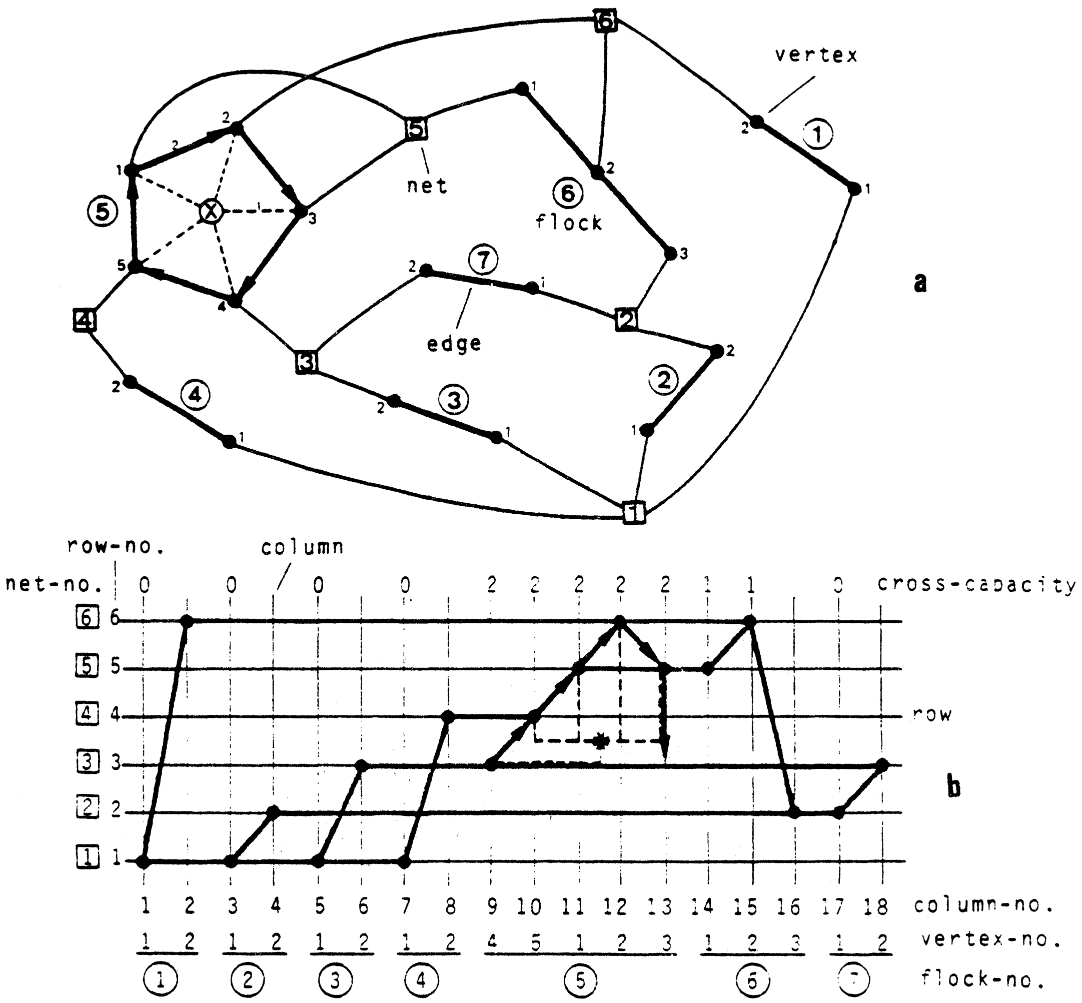


FIGURE 2 Grid-embedding of a circuit graph S a) into a planarization graph G b) with natural assignment. The solution is infeasible.

introduce a neighbour-list $NN(j, k) = (N_1, N_2, ..., N_j)$ which characterizes a net-arrangement uniquely in the following way: Each net-triple $...N_a, N_b, N_c...$ within the list prescribes the relative situations of the nets N_a, N_b, and N_c in the grid-model in the way that N_a must lie under N_b and N_b must lie under N_c. Here j is the number of nets in the list and k is the "address" of this arrangement. As an example, for $j = 4$ we have $3! = 6$ different possibilities $NN(4, 1)$ to $NN(4, 6)$, which can be put in order in the way shown in Figure 3.

These six net-arrangements derive from the two with $j = 3$ nets. Figure 3 shows the complete hierarchy, called *solution-tree*, for $n = 4$ nets. Beginning with the root of the tree, which corresponds to the arrangement of only the one net N_1, each net-arrangement and, hence, each solution can be reached by adding net by net to the actual arrangement. Each node of the tree itself represents a root of a subtree. If the root of a subtree is infeasible, every node of this subtree is infeasible, too. This fact helps in exploring the solution-tree implicitly, i.e. it is not necessary to inspect *every* node of the tree, if one can already determine "infeasibility" on a lower level $j < n$.

The net-arrangement-procedure works in principle as follows: Starting with the root on level $j = 1$ a depth-first-search on the solution-tree is performed, looking for feasible nodes. If an infeasible node is

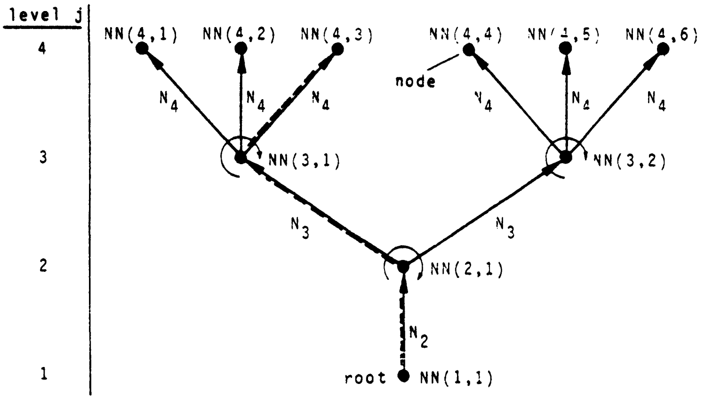


FIGURE 3 Complete solution tree for $n = 4$ nets.

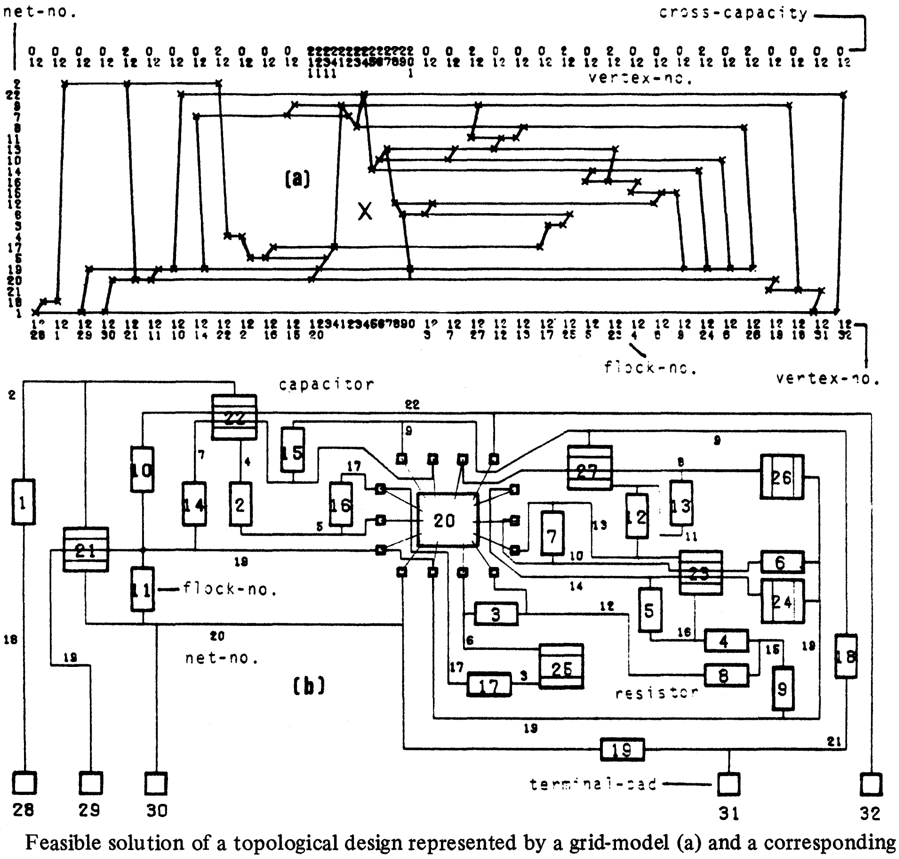


FIGURE 4 Feasible solution of a topological design represented by a grid-model (a) and a corresponding principal layout (b).

attained, a backstep is done. The algorithm ends either with a feasible solution in the highest level or with a backstep to level 0 indicating that no feasible solution exists.

At the beginning and during the search global and local restrictions can be stated, which are based on a fact explained by means of Figure 2b: Net N_6 is living in the column-band bounded by columns 5 and 6. The edge in this band has cross-capacity 0 and is incident to the nets N_1 and N_3. If N_6 is arranged *between* N_1 and N_3, the solution becomes infeasible. Hence, we restrict N_6 *not* to be laid between N_1 and N_3. These simple restrictions lead to a striking improvement of the effectiveness of the net-arrangement procedure.

5. PRACTICAL EXAMPLE

Figure 4 shows the result of a topological design of a circuit with 32 components (incl. 5 boundary terminals) and 22 connection trees. The grid-model in Figure 4a contains 13 feasible crossings. Flock 20 is the image of 4 op-amps integrated in one semiconductor-chip with 14 terminals. The star within the flock indicates the feasible position of the component-body so that each bond-wire will be crossed by at most two conductor lines. The principal routing expressed in the grid-model is realized in the more perceptual representation of a principal layout (Figure 4b), in which uniform sizes of the components are assumed.

REFERENCES

1. P. Dierick, R. Lavie and J.-L. Martin, "Tracé Automatique de Masques de Circuits Intégrés Hybrides", *L'Onde Electrique* 49 (1969) 98–103.
2. K. Zibert and R. Saal, "On Computer aided hybrid circuit layout," *Proc. IEEE Intern. Symp. on Circuits and Systems* (1974) 314–318.
3. H. Beke, S. Mazur, R. Govaerts, W. Sansen and R. van Overstraeten, "CALHYM – A computer program for the automatic layout of large digital hybrid microcircuits", *Proc. European Hybrid Microelectronic Conf. Bad Homburg* (1977) I/1–I/9.
4. N. A. Rose and J. V. Oldfield, "Printed wiring board layout by computer", *Electronics and Power* 17 (1971) 376–379.
5. R. van der Leeden, "A planarization graph model for the computer-aided topological design of hybrid circuit layouts", *Proc. European Conf. on Circuit Theory and Design*, Lausanne, (1978) 513–517.
6. R. van der Leeden, "Ein Verfahren zur Planarisierung von Schaltungsgraphen mit begrenzt kreuzbaren Kanten", *Nachrichtentechnische Zeitschrift* 31 (1978) 889–895.
7. R. van der Leeden, *Zum topologischen Layout-Entwurf hybrider Schichtschaltungen*, Dissertation, TU München, (1979), to be published.
8. A. Basden and K. G. Nichols, "New topological method for laying out printed circuits", *Proc IEE* 120 (1973) 325–328.
9. C. Berge, *Graphs and hypergraphs* (North-Holland Publishing Comp., Amsterdam 1973).