Reprinted from INFORMATION AND CONTROL
All Rights Reserved by Academic Press, New York and London

Vol. 57, Nos. 2–3, May/June 1983
Printed in Belgium

Optimal Orientations of Cells in Slicing Floorplan Designs*

LARRY STOCKMEYER

Computer Science Department, IBM Research Laboratory, San Jose, California 95193

A methodology of VLSI layout described by several authors first determines the relative positions of indivisible pieces, called cells, on the chip. Various optimizations are then performed on this initial layout to minimize some cost measure such as chip area or perimeter. If each cell is a rectangle with given dimensions, one optimization problem is to choose orientations of all the cells to minimize the cost measure. A polynomial time algorithm is given for this optimization problem for layouts of a special type called slicings. However, orientation optimization for more general layouts is shown to be NP-complete (in the strong sense).

1. INTRODUCTION

The increasing complexity of integrated circuits has motivated the development of methodologies for VLSI design that are amenable to automation, if not completely then at least in part. A class of related methodologies, described by Lauther (1980), Otten (1982a, 1982b), Zibert (1974), and Zibert and Saal (1974), among others, first determines the relative positions of basic pieces of the circuit, called *cells*, on the chip. Cells represent parts of the design which are regarded as indivisible, for example, flip-flops or RAMs. The initial placement could be done, for example, by repeated application of a min-cut partitioning algorithm. Various optimizations are then performed on this initial layout to minimize some cost measure such as chip area or perimeter. If each cell is a rectangle with given dimensions, one optimization problem mentioned in Lauther (1980), Otten (1982b), and Zibert and Saal (1974) is to choose orientations of all the cells to minimize the cost measure. In this paper we show that this optimization problem can be solved in polynomial time for layouts of a special type called *slicings* (cf. Otten, 1982a, 1982b). For general layouts, orientation optimization is shown to be NP-complete (in the strong sense).

* Part of this work was done while the author was with the Mathematical Sciences Department, IBM Thomas J. Watson Research Center, Yorktown Heights, New York.

0019-9958/83 $3.00

2. Definitions and Related Work

The given layout consists of an enclosing rectangle subdivided by horizontal and vertical line segments into nonoverlapping rectangles. Two different line segments can meet but cannot cross. See Fig. 1. A rectangle that is not subdivided by a line segment we call a *basic rectangle.* Such a rectangle subdivision is called a *floorplan.* In optimizing the layout we are allowed some freedom in moving line segments and in choosing the dimensions of the basic rectangles. The features of a floorplan F that cannot change are captured in a pair of planar acyclic directed graphs A_F and L_F, each with one source and one sink, and possibly having multiple edges. The graph A_F, the "above-relation," has a vertex for every horizontal segment, including the top and bottom of the enclosing rectangle. For every basic rectangle R, there is an edge e_R directed from segment σ to segment σ', where (part of) σ is the top of R and (part of) σ' is the bottom of R. Thus, there is a one-to-one correspondence between basic rectangles and edges of A_F. The graph L_F, the "left-relation," is defined similarly for vertical segments. A_F and L_F are dual planar graphs. Two floorplans F and G are *equivalent* iff $A_F = A_G$ and $L_F = L_G$. See Fig. 1. Viewing each line segment as a channel, the graphs A_F and L_F fix, for each side of each channel, the order of the basic rectangles which lie on that side.

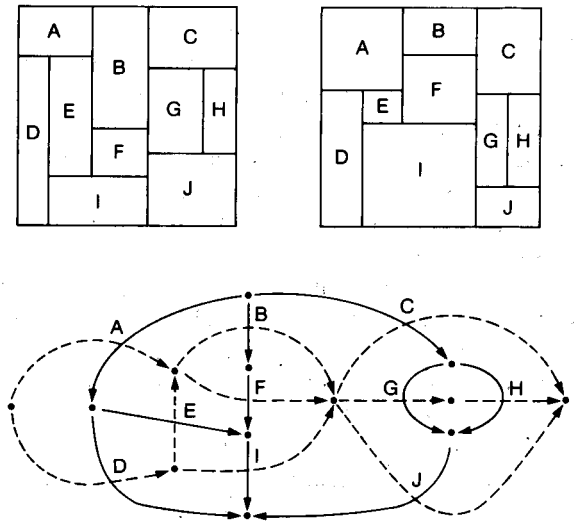

FIG. 1. Two equivalent floorplans and their common above-relation A_F (edges drawn solid) and left-relation L_F (edges drawn dashed). Each basic rectangle in a floorplan corresponds to one edge of A_F and one edge of L_F. When A_F and L_F are embedded as dual planar graphs, these two edges cross.

For the purposes of this paper, a floorplan also associates with each basic rectangle R two positive integers a_R and b_R; these are intuitively the dimensions of a rectangular cell that must fit in R. Each cell has two possible orientations depending on whether the side of length a_R or b_R is horizontal. Given a floorplan F and an orientation ρ of all the cells, there are integers $h_F(\rho)$ and $w_F(\rho)$ giving the minimum height and width, respectively, of a floorplan G equivalent to F for which each cell fits in its associated basic rectangle. (By slight abuse of language we will sometimes identify a cell with its associated basic rectangle.) $h_F(\rho)$ and $w_F(\rho)$ can be defined more precisely as follows. Given the orientation ρ, let $A_F(\rho)$ (resp. $L_F(\rho)$) be obtained from A_F (resp. L_F) by giving each edge e a "length" $l(e)$ equal to the height (resp. width) of the cell corresponding to e. Define an (F, ρ)-*placement* to be a labeling l of the vertices of $A_F(\rho)$ and $L_F(\rho)$ by nonnegative integers such that (i) the sources are labelled zero, and (ii) if e is an edge from vertex σ to vertex σ' then

$$l(\sigma') \geqslant l(\sigma) + l(e).$$

Intuitively, if σ is a horizontal segment, $l(\sigma)$ is the distance from the top of the enclosing rectangle to σ, and the inequality says that the basic rectangle with top σ and bottom σ' is high enough to contain a cell of height $l(e)$, and similarly for vertical segments. Now $h_F(\rho)$ (resp. $w_F(\rho)$) is the minimum label of the sink of $A_F(\rho)$ (resp. $L_F(\rho)$) over all (F, ρ)-placements. Equivalently, it is easy to see that $h_F(\rho)$ ($w_F(\rho)$) is the length of a longest path from the source to the sink in $A_F(\rho)$ ($L_F(\rho)$). We are also given some cost function $\psi(h, w)$, for example, perimeter ($\psi(h, w) = 2h + 2w$) or area ($\psi(h, w) = hw$). The objective is to minimize $\psi(h_F(\rho), w_F(\rho))$ over all orientations ρ.

A floorplan is a *slicing* if either it is a basic rectangle or there is a line segment (a slice) that divides the enclosing rectangle into two pieces such that each piece is a slicing; see Fig. 2. Equivalently, a floorplan is a slicing iff the graphs A_F and L_F are both series-parallel graphs (Otten, 1982). Another useful way to describe a slicing floorplan is to specify the natural hierarchical structure in an oriented rooted binary tree called a *slicing tree*. Each nonleaf node of the tree is labeled either h or v, specifying whether this is a horizontal or vertical slice. Each leaf corresponds to a basic rectangle. See Fig. 2. Requiring slicing trees to be binary simplifies the description of the algorithm in the next section and allows a simple expression for its running time. This requirement causes no loss of generality since a node with $d > 2$ children can be replaced by a binary subtree with d leaves. Thus, there can be several slicing trees that describe a given slicing floorplan. For the algorithm of the next section, a tree of smallest depth should be chosen. Replacing the internal nodes of a tree by binary trees while minimizing depth can be done in time $O(n \log n)$ using an algorithm of Golumbic (1976).

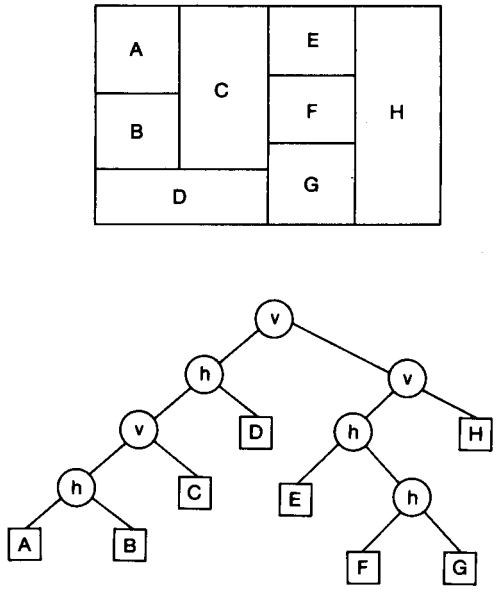

FIG. 2. A slicing floorplan and a slicing tree which describes it.

Otten (1982a, 1982b) has pointed out that slicings have several advantages over nonslicing floorplans, for example, slicings are natural for top–down hierarchical design. Slicings are also used in the design methodology described by Lauther (1980). Although the initial slicing is eventually "squeezed" into a nonslicing floorplan, Lauther suggests doing rotation optimization when the floorplan is still a slicing. He suggests using a greedy heuristic, but we show in the next section that an optimal solution can be found in a reasonable amount of time. Specifically, the time bound is $O(nd)$, where n is the number of cells and d is the depth of the slicing tree. Thus, the time is $O(n^2)$ in the worst case, or $O(n \log n)$ for "balanced" slicings with $d = O(\log n)$. For general floorplans, Zibert and Saal (1974) use integer programming methods to do rotation optimization (as well as several other optimizations simultaneously). Since rotation optimization is *NP*-complete for general floorplans (Section 4) such time consuming methods are probably necessary to find the true optimum in the general case.

3. A Polynomial Time Algorithm for Slicings

THEOREM 1. *Let $\psi(h, w)$ be nondecreasing in both arguments (i.e., if $h \leqslant h'$ and $w \leqslant w'$ then $\psi(h, w) \leqslant \psi(h', w')$) and computable in constant time. For a slicing floorplan F described by a binary slicing tree T, the*

problem of minimizing $\psi(h_F(\rho), w_F(\rho))$ over all orientations ρ can be solved in time $O(nd)$, where n is the number of leaves of T (equivalently, the number of cells of F) and d is the depth of T.

Proof. If u is a node of T, let $F(u)$ denote the floorplan described by the subtree rooted at u and let $L(u)$ be the set of leaves in that subtree. For each node u of T the algorithm constructs a list of pairs

$$(1) \quad \{(h_1, w_1), (h_2, w_2), \ldots, (h_m, w_m)\}$$

with

(1.1) $m \leqslant |L(u)| + 1$,

(1.2) $h_i > h_{i+1}$ and $w_i < w_{i+1}$ for i with $1 \leqslant i < m$,

(1.3) for each i with $1 \leqslant i \leqslant m$ there is an orientation ρ of the cells in $L(u)$ such that

$$(h_i, w_i) = (h_{F(u)}(\rho), w_{F(u)}(\rho)),$$

(1.4) for each orientation ρ of the cells in $L(u)$ there is a pair (h_i, w_i) in the list with

$$h_i \leqslant h_{F(u)}(\rho) \qquad \text{and} \qquad w_i \leqslant w_{F(u)}(\rho).$$

The meaning of (1.4) is that we do not keep $(h(\rho), w(\rho))$ in the list if there is another orientation ρ' that is strictly better than ρ in the h or w dimension (or both) and is not worse than ρ in either dimension. Since the cost function ψ is nondecreasing, we can minimize ψ over all orientations by minimizing $\psi(h_i, w_i)$ over all pairs (h_i, w_i) in the list constructed for the root of T. In order to construct an orientation that minimizes ψ, the algorithm also keeps two pointers with each pair in each list.

The algorithm begins by constructing a list (1) for each leaf of T. If the leaf cell has dimensions a and b with $a > b$, the list is $\{(a, b), (b, a)\}$ and the pointers are null. If $a = b$, there is just one pair (a, b) in the list. (If for some reason the cell has a fixed orientation, then the list has one pair defined by the fixed orientation.) Note that in this case there is one leaf in the subtree and the length of the list is $\leqslant 2$ as required by (1.1). The algorithm now works its way up the tree. In general, let u be a nonleaf node of T with children v and v', say that u specifies a vertical slice, and let

$$\{(h_1, w_1), \ldots, (h_k, w_k)\},$$
$$\{(h'_1, w'_1), \ldots, (h'_m, w'_m)\},$$

be the lists constructed for v and v', respectively, where

$$k \leqslant |L(v)| + 1 \qquad \text{and} \qquad m \leqslant |L(v')| + 1.$$

Note that a pair (h_i, w_i) and a pair (h'_j, w'_j) can be put together to give a pair

$$\text{join}((h_i, w_i), (h'_j, w'_j)) = (\max(h_i, h'_j), w_i + w'_j)$$

in the list for u (see Fig. 3). A key fact is that we do not have to consider all km such new pairs since many of them are clearly suboptimal. For example, in Fig. 3 with $h_i > h'_j$, there is no reason to join (h_i, w_i) with (h'_z, w'_z) for any $z > j$ since, recalling (1.2),

$$\max(h_i, h'_z) = \max(h_i, h'_j) = h_i,$$
$$w_i + w'_z > w_i + w'_j.$$

The following procedure for combining the two lists to obtain a list for u is similar to merging two sorted lists.

(1) Initialize $i \leftarrow 1, j \leftarrow 1$.

(2) If $i > k$ or $j > m$ then halt.

(3) Add $\text{join}((h_i, w_i), (h'_j, w'_j))$ to the list for u with pointers to (h_i, w_i) and (h'_j, w'_j).

(4) If $h_i > h'_j$ then $i \leftarrow i + 1$ and go to 2.

(5) If $h_i < h'_j$ then $j \leftarrow j + 1$ and go to 2.

(6) If $h_i = h'_j$ then $i \leftarrow i + 1, j \leftarrow j + 1$, and go to 2.

Note that the running time of this procedure is $O(k + m)$. The length of the list produced for u is at most

$$k + m - 1 \leqslant |L(v)| + 1 + |L(v')| + 1 - 1 = |L(u)| + 1$$

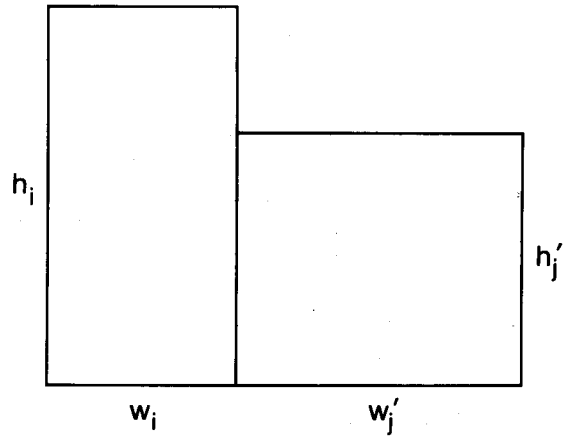

FIG. 3. Illustrating the merging step in the algorithm of Theorem 1.

so (1.1) is satisfied for the new list. The reader can easily check that (1.2)–(1.4) are also satisfied, assuming by induction that they are satisfied for the two lists for v and v'. It is obvious how to modify this procedure if u specifies a horizontal slice. After the list (1) for the root of T is constructed and ψ is minimized over all pairs in the list, it should be clear how to use the pointers to reconstruct an orientation that achieves the minimum ψ.

The running time and storage requirements are both $O(nd)$, where n is the number of leaves (i.e., cells) of T and d is the depth of T. To see this, simply note that for each fixed k with $0 \leqslant k \leqslant d$, the sum of the lengths of the lists constructed for all nodes at depth k is at most $2n$, and the time to construct all of these lists in $O(n)$. ∎

4. *NP*-Completeness for General Floorplans

We assume familiarity with the definition of *NP*-completeness in the strong sense (Garey and Johnson, 1979). Briefly, this means that the problem is *NP*-complete when restricted to instances where the magnitudes of all numbers appearing in the instance are bounded above by some fixed polynomial in the size of the instance. Orientation optimization is defined as a recognition problem in the usual way.

ψ-OPTIMAL ORIENTATION

Instance. A floorplan F and an integer k.

Question. Is there an orientation ρ of the cells of F such that $\psi(h_F(\rho), w_F(\rho)) \leqslant k$?

THEOREM 2. *Let $\psi(h, w)$ be strictly increasing in both arguments (i.e., $h < h'$ implies $\psi(h, w) < \psi(h', w)$ and $w < w'$ implies $\psi(h, w) < \psi(h, w')$) and computable in polynomial time. Then* ψ-OPTIMAL ORIENTATION *is NP-complete in the strong sense.*

Proof. The problem clearly belongs to *NP*. We show that the following 3-PARTITION problem is reducible to ψ-OPTIMAL ORIENTATION. 3-PARTITION is known to be *NP*-complete in the strong sense (Garey and Johnson, 1979).

3-PARTITION

Instance. A set $\{a_1,..., a_n\}$ of n positive integers and positive integers B and m such that $a_1 + \cdots + a_n = mB$.

Question. Can $\{1, 2,..., n\}$ be partitioned into m pairwise disjoint sets $I_1,..., I_m$ such that, for $1 \leqslant i \leqslant m$, $\sum_{j \in I_i} a_j = B$?

(The definition of 3-PARTITION in Garey and Johnson (1979) has the restrictions $n = 3m$ and $B/4 < a_j < B/2$ for all j, but these are not essential to our reduction.)

Consider the floorplan shown in Fig. 4. Define integers D_j, E_j $(1 \leqslant j \leqslant n)$, C, D, H, W as follows:

$$
\begin{aligned}
&D_n = m - 1, \\
&D_j = D_{j+1} + (m-1)(a_{j+1} + 1) \qquad \text{for} \quad 1 \leqslant j < n, \\
&D = D_1 + D_2 + \cdots + D_n, \\
&W = D + B, \\
&C = W + 1 + mD_1/(m-1) + \max_j a_j, \\
&E_j = C - mD_j/(m-1) - a_j, \\
&H = (m-1)\,C.
\end{aligned}
$$

For each i with $1 \leqslant i \leqslant m$ and each j with $1 \leqslant j \leqslant n$, cell X_{ij} has dimensions D_j by $D_j + a_j$ and cell Y_{ij} has dimensions E_j by 1. Cell U has dimensions H by H, and cell V has dimensions $H + W$ by $H + W$. All the small unlabeled cells in Fig. 4 have dimensions 1 by 1.

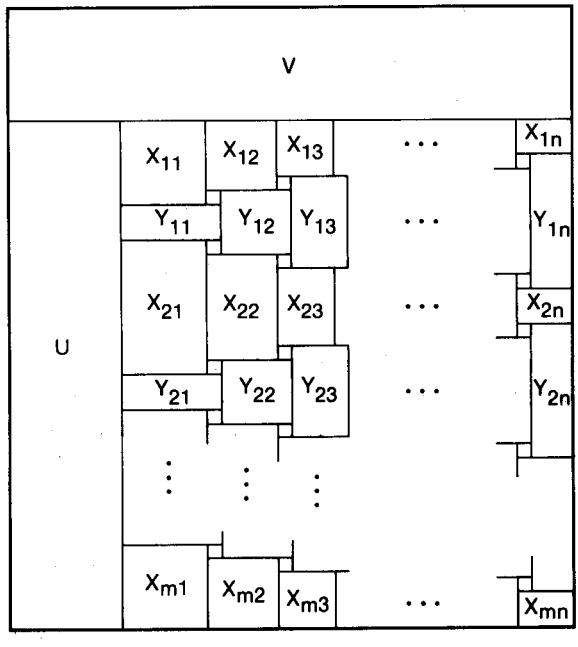

FIG. 4. The floorplan used in the proof of Theorem 2.

Let $k = \psi(2H + W, H + W)$. Let F be the floorplan of Fig. 4 with cells U and V deleted. Since ψ is strictly increasing, it is easy to see that there is an orientation of all the cells in Fig. 4 that gives cost k iff there is an orientation ρ of the cells of F with $h_F(\rho) \leqslant H$ and $w_F(\rho) \leqslant W$. We show that there is such a ρ iff the partition problem has a solution.

Only if Let ρ be an orientation of F with $h_F(\rho) \leqslant H$ and $w_F(\rho) \leqslant W$. Define a solution $I_1,\ldots, I_m$ to the partition problem as follows. For each i with $1 \leqslant i \leqslant m$ and j with $1 \leqslant j \leqslant n$,

$$j \in I_i \text{ iff } \rho \text{ orients cell } X_{ij} \text{ so that it has width } D_j + a_j \text{ and height } D_j.$$

Note that since $E_j > W$, ρ must orient Y_{ij} so that it has height E_j and width 1. For each i, by considering the path from the left side of F to the right side that passes through $X_{i1}, X_{i2},\ldots, X_{in}$, one sees that

$$\sum_{j \in I_i} a_j \leqslant B.$$

For each j, by considering the path from the top to the bottom that passes through $X_{1j}, Y_{1j}, X_{2j}, Y_{2j},\ldots, Y_{m-1,j}, X_{mj}$, it follows that j appears in at least one I_i; otherwise the length of this path would be

$$(m-1)E_j + m(D_j + a_j) = H + a_j > H.$$

Since the sum of the a_j's is mB, it follows that each j appears in exactly one I_i, and that

$$\sum_{j \in I_i} a_j = B \qquad \text{for all } i.$$

if Let $I_1,\ldots, I_m$ be a solution to the partition problem. For each i and j, let ρ orient X_{ij} to have width $D_j + a_j$ iff $j \in I_i$. Let ρ orient all Y_{ij} to have height E_j. We show that there is an (F, ρ)-placement that witnesses $h_F(\rho) = H$ and $w_F(\rho) = W$. Consider first $h_F(\rho)$. Define a path in $A_F(\rho)$ to be *critical* if there exists a j such that each edge in the path is associated with either X_{ij} or Y_{ij} for some i. For every horizontal segment σ of F other than the top and bottom of F, define $l(\sigma)$ to be the length of the unique critical path from the source (top) to σ in $A_F(\rho)$. There are n critical paths from the source (top) to the sink (bottom), but the jth such path has length

$$(m-1)E_j + (m-1)(D_j + a_j) + D_j = H$$

independent of j, so the sink is labeled H. The inequality

$$l(\text{head}(e)) \geqslant l(\text{tail}(e)) + l(e)$$

is clearly satisfied for each edge e associated with an X_{ij} or Y_{ij}. We must check that it is satisfied for the small cells of size 1 by 1. That is, letting t_{ij} (b_{ij}) denote the label of the top (bottom) of Y_{ij}, it must be checked that

$$t_{ij} \geqslant t_{i,j+1} + 1$$

and

$$b_{i,j+1} \geqslant b_{ij} + 1 \qquad \text{for all} \quad 1 \leqslant i < m \text{ and } 1 \leqslant j < n.$$

By straightforward calculations that are left to the reader, these inequalities follow from the definition of the D_j and E_j and the obvious inequalities

$$\begin{aligned} t_{ij} &\geqslant (i-1)E_j + (i-1)(D_j + a_j) + D_j, \\ t_{i,j+1} &\leqslant (i-1)E_{j+1} + i(D_{j+1} + a_{j+1}), \\ b_{i,j+1} &\geqslant iE_{j+1} + (i-1)(D_{j+1} + a_{j+1}) + D_{j+1}, \\ b_{ij} &\leqslant iE_j + i(D_j + a_j). \end{aligned}$$

The argument that $w_F(\rho) = W$ is easier. The placement of the left and right sides of the X_{ij} are determined exactly by the widths of these cells (which are determined by the solution to the partition problem as described above). Since the Y_{ij} all have width 1, there is considerable freedom in placing the left and right sides of these cells. Details of the placement are left to the reader. ∎

5. Conclusion

By isolating the orientation problem it has been possible to define the problem precisely and show that it is tractable for slicings although it is *NP*-complete in general. However, a more realistic model must consider the orientation problem in the context of the full design process, including initial placement, orientation, and wire routing. For example, one would expect the area for wires to depend on the particular orientation. A problem for future work is to model and investigate the complexity of orientation and wire routing together.

Acknowledgments

I am grateful to Ralph Otten for bringing the orientation problem to my attention and for several helpful discussions. I also thank Maria Klawe for a useful conversation concerning the proof of Theorem 2.

Received: February 8, 1983

REFERENCES

GAREY, M. R., AND JOHNSON, D. S. (1979), "Computers and Intractability—A Guide to the Theory of *NP*-Completeness," Freeman, San Francisco, 1979.

GOLUMBIC, M.C. (1976), Combinatorial merging, *IEEE Trans. Comput.* **C-25**, 1164–1167.

LAUTHER, U. (1980), A min-cut placement algorithm for general cell assemblies based on a graph representation, *J. Digital Systems* **4**, 21–34.

OTTEN, R. H. J. M. (1982), Automatic floorplan design, *in* "Proceedings, 19th Design Automation Conf.," pp. 261–267. (a)

OTTEN, R. H. J. M. (1982), Layout structures, *in* "Proceedings, IEEE Large Scale Systems Symp." (b)

ZIBERT, K. (1974), "Ein beitrag zum topologischen entwurf von hybridschaltungen," Doctoral thesis, Techn. Univ. Munich.

ZIBERT, K., AND SAAL, R. (1974), On computer aided hybrid circuit layout, *in* "Proceedings, 1974 IEEE Intl. Symp. on Circuits and Systems," San Francisco, pp. 314–318.

Printed by the St. Catherine Press Ltd., Tempelhof 41, Bruges, Belgium