AN ALGORITHM FOR FINDING A RECTANGULAR DUAL OF A PLANAR GRAPH FOR USE IN AREA PLANNING FOR VLSI INTEGRATED CIRCUITS.

Krzysztof Kozminski and Edwin Kinnen

Department of Electrical Engineering,

The University of Rochester, Rochester, NY 14627.

Abstract.

An $O(n^2)$ algorithm for finding a rectangular dual of a planar triangulated graph is presented. In practice, almost linear running times have been observed. The algorithm is useful for solving area planning problems in VLSI IC design.

Introduction.

Recent work in circuit layout for VLSI has increased interest in the problem of finding a rectangular dissection that satisfies specified neighborhood relations among the component rectangles and results in the smallest area, given a set of constraints on these rectangles[1,2,3]. Earlier, a similar problem was investigated for its application to architectural design[4,5]. In VLSI IC design, this problem arises during area allocation for subsequent layout of the functional blocks of the IC[1,2]. Initially, the entire circuit is partitioned into clusters which can be represented as vertices of a graph. Edges of this graph represent interconnections that carry signals among the clusters. The cluster interconnection graph is planarized and a dual graph is found where faces correspond to the clusters and the edges are the interfaces between clusters through which the intercluster connections pass. A drawing of the dual graph is sought such that the area allocated to each circuit cluster is a rectangle. The dimensions are adjusted to provide (1) sufficient interfaces between clusters to route the corresponding connections, and (2) an area large enough to contain the layout of the corresponding cluster. The layout area is minimized subject to these constraints.

This paper is concerned with a subproblem that has to be solved for these applications: *determine if there exists a rectangular dual for a given cluster interconnection graph.* A theoretical characterization of graphs with rectangular duals has been reported[4], however, an efficient algorithmic implementation of this theory appears impossible[4,5]. The method for rectangular dualization presented in [2] does not appear to be of polynomial complexity. Recently, necessary and sufficient conditions having an efficient algorithmic implementation were developed when nodes in the rectangular dual are constrained to have a degree at most three[6]. This constraint is believed to have little importance in applications[3,7].

Basic theory [6].

For a rectangular dissection D, a rectangle adjacency graph $ID(D)$ is created by assigning a vertex to each rectangle and joining two vertices with an edge whenever the corresponding rectangles abut. $ID(D)$ and D bear dual relationships.

A **rectangular dual** of a graph G is any rectangular dissection D where $ID(D)$ and G are isomorphic. A rectangular dual may not be unique.

The **outermost cycle** of a **plane** graph (a planar graph properly drawn in a plane) with no cut vertices is the cycle with the property that all edges of this graph lie either on this cycle or in its interior. The vertices on the outermost cycle are called **outer**; all others are **internal vertices.**

A **4-completion** of a graph G is any plane graph H such that the graph H' obtained from H by deleting the outer vertices of H together with the incident edges is isomorphic to G and the following conditions hold: (a) H has exactly 4 outer vertices of degree ≥ 3, (b) all internal vertices of H have degree ≥ 4, (c) all faces of H have degree ≤ 4, (d) all faces containing an outer vertex have degree 3 and (e) all cycles in H that are not faces have length ≥ 4. The outer vertices of a 4-completion have a simple interpretation: they correspond to the upper, left, bottom and right sides of the outermost cycle of a rectangular dual of G. There may be several different 4-completions for a given graph G.

THEOREM 1: A plane graph G with all faces triangular has a rectangular dual iff there exists a 4-completion of G. □

To construct a 4-completion of a plane graph **G** one creates a cycle of 4 additional vertices u, l, b, r containing G in its interior. Another 4 vertices are picked on the outermost cycle of G. These vertices are called corner vertices as they will map into the corner faces of the rectangular dual; they divide the outermost cycle of G into 4 edge-disjoint paths. All vertices on each of these paths are connected with an edge to one of the added vertices, as shown on fig. 1.

A **shortcut** in a plane **block** (a graph with no cut vertices) G is an edge that has both vertices on the outermost cycle and does not belong to the outermost cycle; see fig. 2.

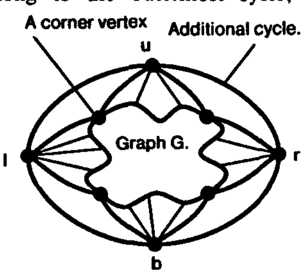
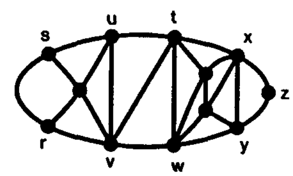


Figure 1.



Shortcuts: uv, vt, tw, xy.
Critical shortcuts: uv, xy.
Corner implying paths: usrv, xzy.

Figure 2.

A uv path taken on the outermost cycle of a plane graph G is said to **imply a corner** if vertices u, v are endpoints of a uv shortcut and if this path contains no other endpoints of a shortcut; this shortcut is said to be **critical**. Corner implying paths may share endpoints but the paths are edge disjoint. They always contain at least 3 vertices, as a single edge would create a face of length 2 with a shortcut.

The endpoints of a shortcut (e.g., u and v in fig. 2) cannot be connected to the same vertex of the 4-completion under construction as this would create a triangle not being a face, thus violating condition (e) above. Therefore, the corner implying path from u to v must contain one of the corner vertices.

THEOREM 2: A plane block G that (a) has all faces of degree 3, (b) has all internal vertices with degree at least 4, and (c) all cycles in G that are not faces have degree ≥ 4, has a rectangular dual iff it has no more than 4 corner implying paths. □

In general, G may contain several maximal blocks. A **block neighborhood graph (BNG)** for a given graph G is a graph where each vertex corresponds to a maximal block of G, and two vertices are connected with an edge iff the corresponding blocks have a vertex in common. A **critical corner implying path** in a maximal block G_i of a plane graph G is a corner implying path of G_i that does not contain cut-vertices of G.

THEOREM 3: A plane graph G that fulfills the assumptions from theorem 2 except for being a block, has a rectangular dual iff (1) its BNG is a path, (2) end-blocks corresponding to the endpoints of the BNG each contain at most 2 critical corner implying paths, and (3) no other maximal block contains a critical corner implying path. □

The algorithm.

An algorithm for finding a rectangular dual of a plane graph G has been developed and implemented. The first 3 steps are numerical checks done in linear time. Failure of G to comply with these checks causes the algorithm to terminate as no 4-completion exists for G.
Step 1: Check all internal vertices of G for degree >4 and all faces for degree 3.
Step 2: Find all maximal blocks of G using depth first search. Check if the BNG, obtained as a byproduct, is a path.
Step 3: Identify shortcuts and critical corner implying paths in G. Proceed further only if there are at most 4 critical corner implying paths and, if G is not a block, if they occur in the end-blocks of G, at most 2 per end-block.
Step 4: Pick any vertex from the interior of each of the critical corner implying paths. If $n<4$ vertices were chosen, select any $4-n$ vertices of the outermost cycle of G so that this cycle is divided into 4 edge disjoint paths. The chosen vertices are the corner vertices. Construct a 4-completion H as shown in fig. 1.
Step 5: This step assigns vertical and horizontal orientations to the edges of the rectangular dual of G and is the most complex one in the algorithm. The orientations of the edges in the dual graph are the only missing information that is necessary to create a rectangular embedding; generating an incidence matrix of a dual graph from H is straightforward. Step 5 divides the 4-completion into smaller 4-completions. This corresponds to cutting the (as yet unknown) rectangular embedding into parts having a smaller number of faces; these parts are then merged along the cut lines. Step 5 can be described in a recursive manner:
Case 0. There exists an edge ub or lr. The original graph G contained a triangle that was not a face. Terminate the algorithm.
Case 1. H is a trivial 4-completion consisting of 5 vertices and 8 edges, as shown in fig. 3. Edges dual to uv and bv are made horizontal, edges dual to lv and rv are vertical.

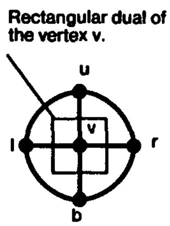


Figure 3. Trivial 4-completion.

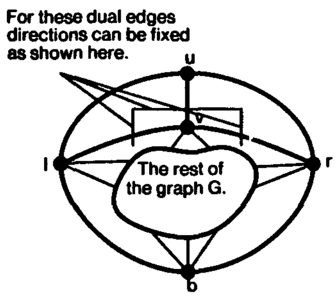


Figure 4.

Case 2. Some outer vertex of H has degree 3; if we assume this is vertex u then H must appear as in fig. 4. Make edges dual to lv and vr vertical and the edge dual to vu horizontal. Delete vertex u together with the corresponding edges and obtain graph H'. If H was a 4-completion, then so is H', for all properties (b)-(e) are preseved if H is not trivial. Property (a) must be also preserved for if some outer vertex, say r, acquires degree 2 in H' then triangles rvb and lvb would exist and violate condition (e) in H. Apply step 5 of the procedure to H'.
Case 3. All outer vertices of H have degree 4. H must appear as in fig. 5a. From the triangularity of faces of H, there exists a vertex x adjacent to l and b and a vertex y adjacent to u and r. Find a shortest path between u and b not passing through r, l, x or y. If such a path does not exist, then based on the triangularity of faces in G, edge xy must exist; take path $uxyb$ instead. This path is called a **cutting path**. It separates 2 internal vertices of H, thereby insuring that the parts of H will have fewer vertices than H. Assign a

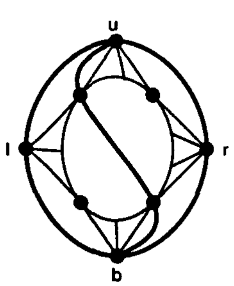


Figure 5a.

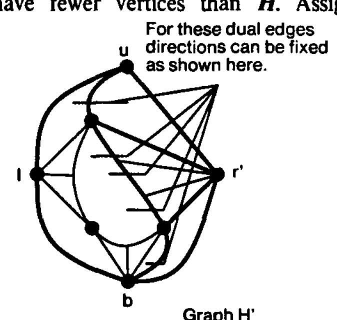


Figure 5b.

horizontal orientation to all edges dual to the edges of the cutting path. Construct two graphs H' and H'' by splitting H along the cutting path and adding vertices l', r' with edges to all the vertices of the cutting path, as shown in fig. 5b for H'. H' and H'' are 4-completions, as the construction preserves all properties of a 4-completion except possibly (c) and (e); if (c) or (e) is violated during the construction, as H has triangular faces, the cutting path can be shown not to be a shortest one. Apply step 5 to H' and H''.
Step 6: Based on the known incidence information and orientations of the dual edges, a rectangular dual is drawn.

The computational complexity of the first 4 steps is linear in the number of edges, as it is only required to check the degrees of the faces and vertices of G and to scan the list of edges to determine shortcuts. A clockwise scan of the vertices on the outermost cycle of each maximal block of G uncovers the critical shortcuts and corner implying paths.

As each application of a cutting path to split H results in at least one edge of the dual graph being assigned an orientation, the number of times the splitting algorithm is applied for step 5 is linear in the number of edges of the graph G. A Lee algorithm that has complexity linear in the number of edges of the graph is used to find a shortest cutting path. Thus the total complexity of the step 5 is at most $O(e^2)=O(n^2)$, as the numbers of edges and vertices in a triangulated graph are linearly related. Step 6 can be performed $O(e)=O(n)$ time.

An example of rectangular dualization.

Fig. 6 shows a 4-completion of a graph for which a rectangular dual is sought. Application of the method described in [2] would result in a system of 110 equations with 216 unknowns to be solved in integers. The generation of the rectangular dual using the algorithm described here took 1 second CPU time of a VAX 750. The result is plotted in fig. 7.

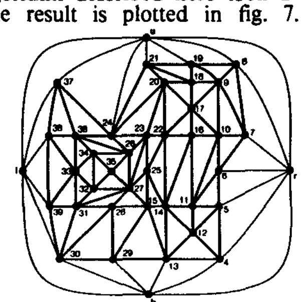


Figure 6.

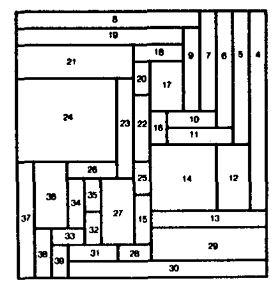


Figure 7.

Conclusions.

A method for the efficient construction of a rectangular graph has been presented together with the theorems stating necessary and sufficient conditions of an existence of rectangular duals for a class of graphs. A method for efficient enumeration without repetitions of all rectangular duals with the running time limited by the number of generated duals is currently being implemented.

Acknowledgments.

The authors thank Professor Arthur H. Stone for his helpful comments and discussion. This work has been supported in part by the Semiconductor Research Corporation, Research Triangle Park, North Carolina, and Scientific Calculations, Inc., Fishers, New York.

References.

[1] Heller W.R., Sorkin G., Maling K., The Planar Package Planner for System Designers, *Proc. of the 19th DAC, IEEE and ACM*, Las Vegas, June 1982, pp. 253-260.
[2] Maling K., Mueller S.H., Heller W.H., On Finding Most Optimal Rectangular Package Plans, *Proc. of the 19th DAC, IEEE and ACM*, Las Vegas, June 1982, pp. 663-670.
[3] Brebner G., Buchanan D., On compiling structural descriptions to floorplans., Proc of the IEEE ICCAD, Santa Clara, Sep12-15 1983, pp 6-7.
[4] Grason J., A dual linear graph representation for space filling location problems of the floor plan type, *Emerging methods in Environmental Design and Planning*. G.T.Moore, (Ed). *Proc. of The Design Methods Group, 1st International Conf.*, Cambridge, MA., 1968, pp. 170-178.
[5] Steadman P., Graph-theoretic representation of architectural arrangement. *The Architecture of Form*. L. March, (Ed) Cambridge Univ. Press. London, New York, Melbourne, 1976, 94-115.
[6] Kozminski, K.A., Kinnen E., Rectangular duals of planar graphs, to be published in *Networks*.
[7] Dr Peter Bruell, Texas Instruments Inc., private communication.