Systolic algorithms for rectilinear polygons

Rajiv Kane* and Sartaj Sahni

We develop systolic algorithms for the OR, AND, Oversizing, and Undersizing of rectilinear polygons. These algorithms work on an edge representation of the polygons rather than on a bit map representation. The algorithms are to be run on a systolic chain of processors. The edges are input at the left end of this chain. From here, they 'float' as far to the right as necessary. As edges float to the right, they compare themselves with edges that are resident in the processors they are floating through. During this comparison the output polygons are generated. Output polygons float to the left. These polygons are output from the left end of the chain. The throughput of the sytolic system can be improved by increasing the length of the processor chain.

rectilinear polygons, systolic algorithms, oversize, undersize, circuit design

Rapid advances in manufacturing technology have made it possible to fabricate chips of ever-increasing complexity. This has posed a severe challenge to existing design automation tools. Existing algorithms take more computer time than is desirable and in some cases require more time and memory than is practical.

One way to meet this challenge is to design new computer architecture and corresponding algorithms for design automation tasks. This approach has been the subject of many recent research efforts. Special architectures for design rule checks are described in Blank *et al*[1], Seiler[2] and Kane and Sahni[3], wire routing is considered in Blank *et al*[1], Mudge *et al*[4], Nair *et al*[5], and Ueda *et al*[6], Iosupovici *et al*[7] and Dah-Juh and Breuer[8] consider module placement. New architectures for simulation are proposed in Abramovici *et al*[9], Denneau[10], Kronstadt and Pfister[11], and Pfister[12]. It is anticipated that through the use of these specialized architectures, one can increase the circuit size that can be handled by a few orders of magnitude.

Kane and Sahni[3] have proposed a systolic design rule checker. This is essentially a hardware algorithm that checks for width and spacing errors. (The reader unfamiliar with systolic designs is referred to Kung[13] for an excellent introduction.) This paper is a continuation of the work reported in Kane[3]. Specifically, systolic algorithms are developed for some functions that are commonly performed on rectilinear polygons. These functions are:

- OR
  - find the logical OR of a set of rectilinear polygons belonging to k layers
- AND
  - find logical AND of a set of rectilinear polygons belonging to k layers
- UNDERSIZE
  - reduce the height (width) of each rectilinear polygon by a specified amount $2d$
- OVERSIZE
  - increase the height (width) of each polygon by a specified amount $2d$

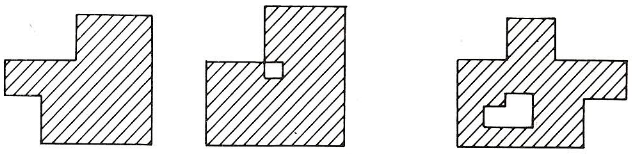
The details of these functions are described in the next section. The systolic algorithms described in this paper are edge based. This coupled with the fact that the algorithms are to be implemented in hardware should result in a substantial reduction in the computing time required.

RECTILINEAR POLYGONS AND FUNCTIONS

In this paper only rectilinear polygons are dealt with explicitly. A rectilinear polygon is composed solely of horizontal and vertical edges. Further, it is assumed that all polygons are well formed. This means that open polygons and polygons with self overlaps (Figures 1(a) and (b)) are not permitted. However, polygons are permitted to contain holes which are themselves rectilinear polygons (Figure 1(c)).

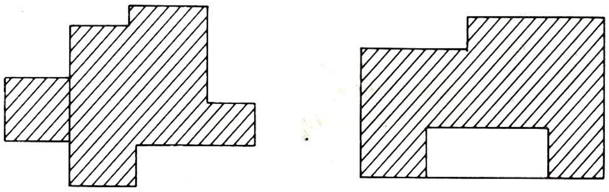
The set of polygons to be handled is assumed to belong to different layers. It is assumed that a polygon belonging to a given layer satisfies the spacing and width requirements as described in Kane[3]. These requirements can actually be relaxed for the AND, and OR operations. In particular, polygons on a layer can share an edge, and a hole may share an edge with the enclosing polygon (see Figure 2). However, these are not allowed for oversize and undersize operations.

The restriction to rectilinear polygons allows a compact



a b c

Figure 1. Polygons (a) open, (b) with self overlap and (c) with hole that is a rectilinear polygon. Polygon interior is shaded



a b

Figure 2. Polygons (a) on a layer sharing an edge and (b) enclosing and sharing an edge with a hole. Polygon interior is shaded

Computer Science Dept., 136 Lind Hall, University of Minnesota, Minnesota, MN 55455, USA
*Present address: Daisy Systems, San Jose, CA, USA

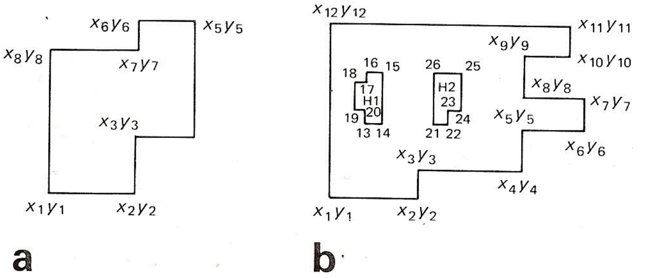


Figure 3. Polygons (a) without holes and (b) with two holes H1 and H2

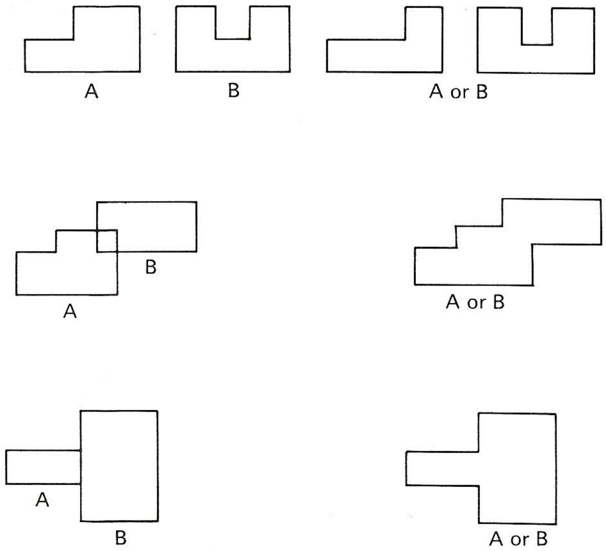


Figure 4. The OR of two rectilinear polygons A and B

representation for each polygon. This representation consists of the following:

- Polygon number
  - Each polygon is assigned a unique number. Holes within a polygon are assigned the same number as the enclosing polygon.
- Layer number
  - The layer number to which the polygon belongs.
- Sequence of polygon vertices

This sequence begins at the lowermost left hand vertex of the polygon and is obtained by traversing the polygon so that its interior lies to the left of the edge being traversed. Since all edges are either horizontal or vertical, the polygon vertices (except the first) may be described by providing a single coordinate. Thus, the polygon of Figure 3(a) is represented as

$$p, n, l, x_1, y_1, x_2, y_3, x_4, y_5, x_6, y_7, x_8, y_1$$

The first symbol p identifies this as an enclosing polygon; n is the polygon number. l is the layer number. In case of a hole, an h is used in place of the p. Holes are traversed such that the interior is to the left of each edge traversed. The representation for the polygon and holes of Figure 3(b) is

$$p, n, l, x_1, y_1, x_2, y_3, x_4, y_5, x_6, y_7, x_8, y_9, x_{10}, y_{11}, x_{12}, y_1$$

$$h, n, l, x_{13}, y_{13}, x_{14}, y_{15}, x_{16}, y_{17}, x_{18}, y_{19}, x_{20}, y_{13}$$

$$h, n, l, x_{21}, y_{21}, x_{22}, y_{23}, x_{24}, y_{25}, x_{26}, y_{21}$$

OR

The OR of two rectilinear polygons A and B is a set of rectilinear polygons that includes the area occupied by A as well as that occupied by B. Figure 4 gives some examples.

AND

The AND of two rectilinear polygons A and B is a third rectilinear polygon C which includes the area that is common to both A and to B. Figure 5 shows some examples.

Oversize

Oversizing in the y direction is described here. Oversizing in the x direction is similar. The objective is to enlarge the polygon by an amount $2d$ in the y direction. For each horizontal edge, with $ud = 0$, the y coordinate is decreased by d and for each horizontal edge with $ud = 1$, the y coordinate is increased by d. The OR of these elongated polygons with the original set (assumed to be a set of non-overlapping polygons) yields a set of polygons oversized by $2d$ in the y direction. Note that the OR is needed to properly account for holes that shrink to zero during the oversize and also to account for the fact that oversizing might cause polygons that did not overlap earlier, to do so now. Figure 6 gives some examples.

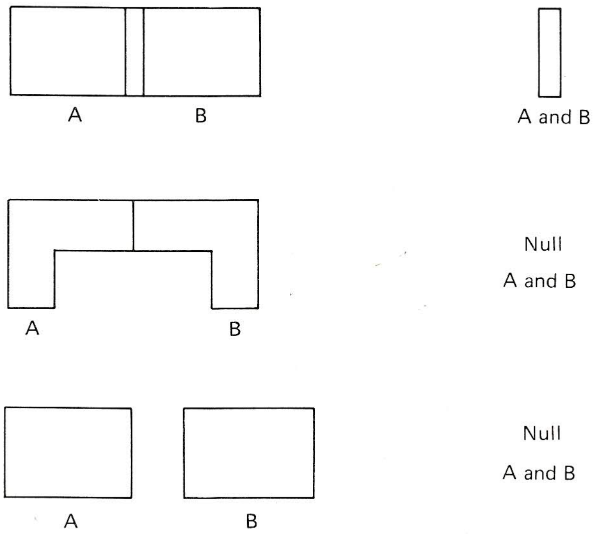


Figure 5. The AND of two rectilinear polygons A and B

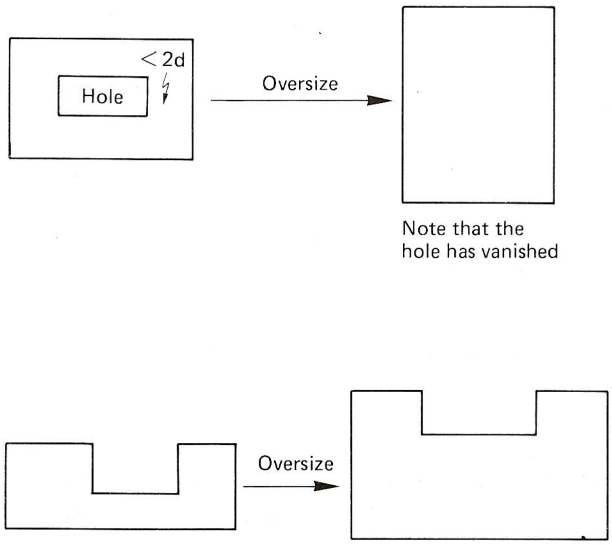


Figure 6. Examples of oversizing