

ALGORITHM OF INPUT DATA PREPARATION FOR SINGLE-BEAM PATTERN GENERATOR IN INDUSTRY-ORIENTED APPLICATIONS

Summary

In this paper, a new algorithm for an automated system of input data preparation for integrated circuit layout generator is proposed. A problem of covering polygons with rectangles is considered. The rectangles must lie entirely within the polygon and it is preferable to cover the polygon with few rectangles as possible. Functions and a structure of the software are described and given some examples of data processing. Results described in this paper can be applied in computational geometry and image analysis in industry-oriented applications.

Key words: covering algorithm, polygon, rectangle, circuit layout generator

ALGORYTM PRZYGOTOWANIA DANYCH WEJŚCIOWYCH DLA GENERATORA POJEDYNCZEJ WIĄZKI OBRAZU W APLIKACJACH PRZEMYSŁOWYCH

Streszczenie

Zaproponowano nowy algorytm dla zautomatyzowanego systemu przygotowania danych wejściowych dla generatora układu scalonego. W pracy rozważany jest problem pokrywania wielokątów prostokątami. Prostokąty muszą zawierać się całkowicie w wielokącie i preferowane jest, aby pokryć wielokąt możliwie niewieloma prostokątami. Funkcje i struktura oprogramowania zostały opisane wraz z przykładami przetwarzania danych. Wyniki opisane w artykule mogą być zastosowane w geometrii obliczeniowej i analizie obrazów w aplikacjach przeznaczonych dla przemysłu.

Słowa kluczowe: algorytm pokrycia, wielokąt, prostokąt, generator układu scalonego

1. Introduction

There is a problem of formation of layout structures on metallized photo masks by manufacture of integrated circuits or printed circuit boards [1]. These structures are formed from type-setting elements with the help of pattern (layout) generators. The type-setting element represents a rectangle. Creation any images of layout structures by means of the generators demands preliminary decomposition of the description of these structures on a set of the rectangles. Thus the number of the rectangles that enclose into the set should be minimal or close to minimal and each rectangle in this set would satisfy to limitations on the sizes (they are determined by technical characteristics of the pattern generators). Special algorithms used for this goal solve a task of a covering of circles, rings, bus-bars, triangles. A universal algorithm solves a task of covering of any polygon given one or several contours. There are a lot of the automated systems [1-11] of input data preparation for IC layout generators; however they are focused on a limited class of layout objects.

2. Aim and scope of the paper

A new heuristic algorithm for an automated system of input data preparation for layout generator is proposed in this paper. Its correctness is assessed on the basis of a set of rectangles determined for covering.

3. Material and methods

A topology is defined using a set of base layout objects: contours, bus-bars and circles. A contour corresponds with

a closed broken (polygonal) that is represented by coordinates of its vertices. A bus-bar is represented by its width and track centerline specifying bus-bar path. A circle is determined by its radius, coordinates of the center and a number determining accuracy of approximation of the circle by a contour. As a rule, the accuracy set additional reduced radius: the rectangles forming the circle have to cover in total all points belonging to the circle with the reduced radius and not to cover any of points outside the determined circle.

The following sequence of steps is carried out during input data preparation for layout generators.

1. The correctness of the set of contours is analyzed. The contour is considered as incorrect if the length of some of its sides is equal to zero or less than minimal length of the side of a rectangle of a type-setting element. The angle between the sides of a contour cannot be equal to zero. Incorrectly given contours are deleted from the set of the contours describing topology.
2. Mutually crossing contours are searched and classed. Contours a and b belong to one class if there are contours c_1, c_2, \dots, c_n in this class such, that pairs $ac_1, c_1c_2, c_2c_3, \dots, c_{n-1}c_n, c_nb$ are crossed among themselves. The contours of one class are replaced with one contour. Afterwards, the description of initial topology is represented as set of mutually non-crossing contours.
3. Simply connected contours and multiply connected contours are allocated.

The simply connected contour does not contain inside itself any other contours. The multiply connected contour consists of the basic contour, which contains inside itself some other contours. The contours included in a multiply

connected contour, refer as contour-cuts. The contour-cut may not be inside any other contour except the basic.

4. A subset of simply connected contours, which are included in a given rectangle, may be inverted. Thus, we suppose, that a rectangle is not contained in the given multiply connected contours. This stage results in the considered subset of simply connected contours replaced with one multiply connected contour in which a rectangle acts as the basic contour, and the contours which are included in the given subset, are contour-cuts.

5. The task of decomposition (covering) is solved for each simply connected and multiply connected contour, i.e. a set of rectangles is found for each of these contours. Then, determined rectangles are jointed in one simply connected contour or multiply connected contour. If the given set of rectangles is a covering of the multiply connected object, then contour-cuts are added also. If this generated union of the rectangles is the simply connected contour, then the received covering is correct. If the resulting contour is multiply connected, the covering is not correct. Non-correct covering is supplemented with the rectangles, and in this way it becomes correct. The obtained sets of rectangles are added in the covering.

6. In the obtained set of the rectangles covering initial layout objects, the optimum sequence of rectangles is found.

Thus, a number of the rectangles enclosed into the result set of rectangles should be minimal or close to minimal, and each rectangle in this set should satisfy its size limitations. A number of intersections between the rectangles can also be minimized and save computation time for finding covering.

So, different tasks are solved during the input data preparation for layout generators, but the most important is layout decomposition or covering by rectangles (Fig. 1).

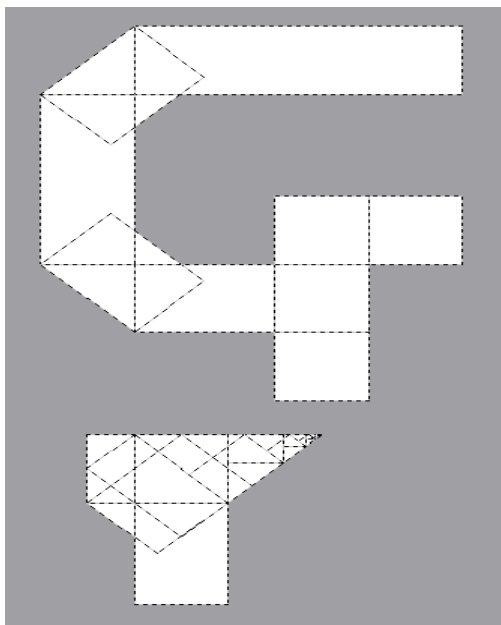


Fig. 1. A result of covering

The covering algorithm finds previous covering, analyses this covering on correctness and finds holes in covering. Then special algorithm covers uncovered holes and finds correct covering or entrance sequence for laser generators.

A line segment is a part of a line that is bounded by two end points, and contains every point on the line between its end points. Examples of line segments include the sides of a triangle or square. In geometry a polygon consists of straight lines that are joined to form a closed chain or circuit. A polygon is a plane figure that is bounded by a closed not crossed itself path, composed of a finite sequence of straight line segments (i.e., by a closed polygonal chain). These segments are called their edges or sides, and the points where two the edges meet are the polygon's vertices or corners. Simple single connected polygon: the boundary of the polygon is single and does not cross itself.

Multiply connected polygon W is traditionally a plane figure that is bounded by a closed polygonal chains: L_1, L_2, \dots, L_g . L_1 is called a main or ground closed polygonal chain and L_2, \dots, L_g closed polygonal chains inside main one L_2, \dots, L_g also called contour-cuts. Multiply connected polygon defines the point of the plane, located on the borders of representing its contours, as well as the point of the plane inside the main closed polygonal chain, but not inside the contour-cuts. Rectangle belongs to a polygon, if any point of the plane, located within or on the border of this rectangle is inside or on the boundary of a multiply connected polygon. The rectangle is h -allowable if the length of any of the parties not less than some value h , where h is a positive real number not equal to zero. The point of the plane r , located inside or on the boundary of the polygon is called the h -covered if there is h -allowable rectangle belongs to this polygon, such that the point r lies on the border or inside the rectangle. In the polygon containing sharp inside corners, there may be a point located near the sharp internal angles, which are not covered by h -allowable rectangles. Moreover, the point of the plane, located at the apex of an acute angle, is not covered by h -allowable rectangles for any quantity.

A covering of a polygon means a set of multiply h -allowable rectangles satisfying the following conditions: 1) every rectangle of a given covering set belongs to multiply connected polygon, 2) for every h -coverable point r from multiply connected polygon there is at least one rectangle such that the point r lies on the border or inside the rectangle. In this paper the following problem is studied: how to find a correct multi connected covering polygon W that consists of a minimum or close to the minimum number of h -allowable rectangles.

The proposed algorithm of covering consists of two stages. In the first stage, its previous covering is generated on the sides of a polygon. In the second stage we evaluate a covering correctness. If the previous covering is not correct, then it is complemented by the rectangles so that the new covering becomes correct. The rectangles are constructed on the base of the sides multiply connected polygon.

Below, for simplicity, polygons will be denoted by Latin letters a, b, c , etc. Let $P(ab)$ be a straight line passing through the side ab of multiply connected polygon.

First stage. Finding previous covering.

The rectangles are constructed on the base of the sides multiply connected polygon. Consistently, get over the sides of polygon. For the every side of polygon we construct one or more rectangles for previous covering by the next specification.

Rule 1. If interior angles on sides of multiply connected polygon are not acute angles, then, the main rectangle is built on this side.

To build a basic rectangle, draw the lines R_1, R_2 perpendicular to the line $P(a)$ and passing through the boundary points of a (Fig. 2). Among the points of the polygon, that lie between these two lines and the side a , we search for point β , which lies on the border of the polygon and located at a minimum distance from side a . Draw a straight line R_3 through this point, R_3 parallel to the line $P(a)$. Intersection points of lines $R_1, R_2, R_3, P(a)$ set vertices of the desired main rectangle S_1 .

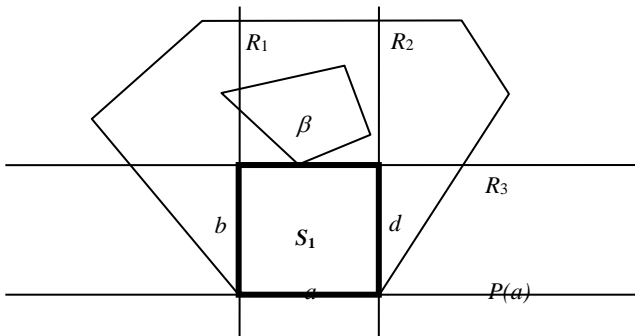


Fig. 2. Previous covering by main rectangle

Further, we extend the main rectangle S_1 to its sides d and b . The basic or main rectangle shown on Fig. 2 cannot be extended to these sides. Therefore, we turn to Fig. 3.

Expansion of the main rectangle: Let p side of the main rectangle S_1 , p is parallel to the side a , and, respectively there are straight lines passing through the sides b, d, p of rectangle S_1 with name R_1, R_2, R_3 . Construct rectangle S_2 on side d of rectangle S_1 . Rectangle S_2 can be constructed only by the side of d if the internal angle on this side of polygon W is greater than or equal to 180 degree. We might find among points of polygon W limited by lines $R_2, R_3, P(a)$, and which is not situated in rectangle S_1 point γ , γ lies on the boundary of a multiply connected polygon, and located at a minimum distance (not equal to zero) from line R_2 . If such a point exists, then draw straight line G_4 through it, G_4 is parallel to line R_2 . Intersections of lines $R_2, R_3, P(a)$ and G_4 define the vertices of new desired rectangle S_2 . Add rectangle S_1 to rectangle S_2 . Similarly, you can try to extend the main rectangle to side b . The main rectangle obtained after extension (S_1+S_2) is included in the previous covering.

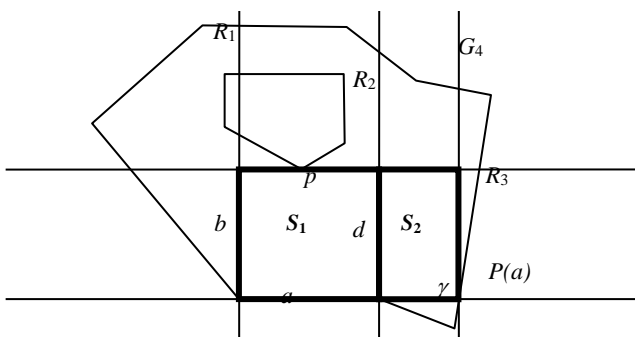


Fig. 3. Previous covering after extension main rectangle by side d

Rule 2. Suppose that one of the interior angles on side a of multiply connected polygon W is an acute in vertex a_1 (Fig. 4). The other end point of side a is denoted a_2 . Draw

line F_1 through boundary vertex a_2 , line F_1 is perpendicular to a . There are two options or variants.

Variant 1. Polygon side t , that is adjacent to side a and connecting with it in vertex a_1 , also intersects with line F_1 . Denote as T and A the lines where sides t and a is located accordingly. Denote F_2 the line, which is the bisector of the angle between polygon sides t and a . Find the point α (α is intersection point of lines F_1 and F_2). Draw line F_3 that is parallel to line A and α also must be situated on this line. Find point β that is the point of intersection lines F_3 and T . Draw the line through this point F_4 , which is parallel to line F_1 .

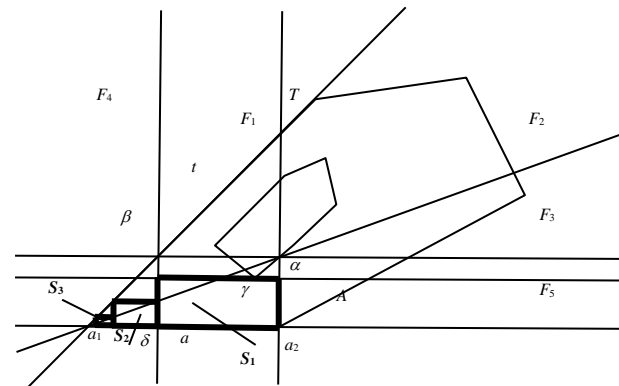


Fig. 4. Construction rectangles for polygon side a , one of polygon interior angles on side a acute (variant 1)

In the set of points of multiply connected polygon, that lie between lines F_1, F_4 and A , find point γ , that is located on the boundary of polygon and on minimum distance from side a . Draw straight line F_5 through this point parallel to line A . If the distance from point γ to polygon side a is less than the distance from α to this side, then points are located at the intersection of lines F_1, F_4, F_5, A for rectangle S_1 (Fig. 4), else points are located at the intersection of lines F_1, F_4, F_3, A . If the length one of the sides for rectangle S_1 is less than the predetermined value h , then the process of generating this rectangle for side a stops because we might built only h -allowable rectangle. If S_1 is h -allowable rectangle then we start a construction for the rectangle S_2 . We built rectangle S_2 just like rectangle S_1 (Fig. 6). δ is the point of intersection for lines F_4, A . As side a put under consideration line, segment is located between the end points a_1 and δ . If the length one of the sides for rectangle S_2 is less than the predetermined value h , then the process of generating this rectangle for line segment $a_1\delta$ of the side a stops because we build only h -allowable rectangle. Otherwise, we add rectangle S_2 in the previous covering, and similarly rectangle S_3 is constructed (Fig. 4), etc.

Variant 2. t is a side of polygon that adjacent to the polygon side a and they are connected in polygon vertex a_1 but t does not intersect with the line F_1 (Fig. 5).

Draw line Z , line Z is perpendicular to the side a and passing through the end point of t , which is not a polygon vertex a_1 . Denote as α point of intersection for lines Z and A , and as a_1, a_2 end points of polygon side a . We split side a into two line segments $(a_2, \alpha), (\alpha, a_1)$. On the base of line segment (a_2, α) construct rectangle for the previous covering as it is described by rule 1. But at the same time as the initial part of the polygon is considered segment (a_2, α) .

Accordingly variant 1 of rule 2 on base of line segment (α, a_1) as a part of polygon side a construct some rectangles for the previous covering. At the same time in this case (Fig. 5), can be constructed several rectangles for previous covering.

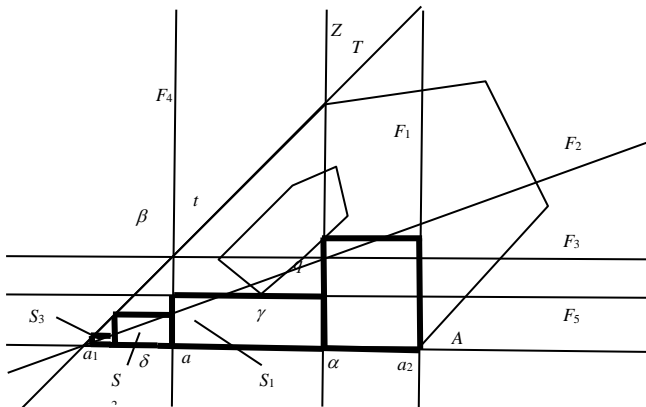


Fig. 5. Construction rectangles for polygon side a , one of polygon interior angles on side a acute (variant 2)

Rule 3. Assume that both internal angles of the multiply connected polygon, on side a with end points a_1, a_2 are an acute. In this case, find a point β located in the middle of side a . Thus, we split side a into two segments (a_1, β) and (β, a_2). For each of these segments (a_1, β) and (β, a_2) of the polygon side find previous covering as it is described in Rule 2.

Analysis of the previous covering on correctness.
The above algorithm for solving the covering problem is heuristic. It's mean that we can't guarantee a valid solution of the covering task. A multiply connected polygon and its previous covering (it is incorrect, since it contains two non-covered holes) is shown on Fig. 6.

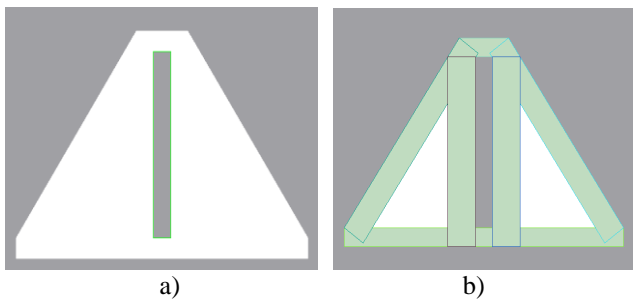


Fig. 6. Multiply connected polygon: a) input polygon; b) covering

For the every side of polygon we construct one or more rectangles for previous covering. Feature of the previous covering proposed above is that to each side of a multiply connected polygon adjoins one or more rectangles from covering. Exceptions are small areas in sharp internal corners and new contour-cuts added after previous covering. This areas and contour-cuts are not covered at the first stage.
Second stage. Analysis of previous covering on correctness and cover added at first stage contour-cuts (Fig. 7).

4. Results

The developed algorithm was implemented in an automated system of input data preparation for integrated circuit

(IC) layout generator. Input data for the system are the description of IC layout presented in GDS II format [12] or in internal text format. It is also necessary to determine the type of generator for which information is being prepared (EM-5009 A2 or EM-5109 [13]), since the accuracy of the final covering for layout patterns is dependent from the corresponding parameters of image generators. Output data of the system are input layout patterns in image format of the generator (formats MUL, BIT, or PAT for layout laser generators EM-5109 and EM-5009A2). The patterns are covered by the rectangles corresponding to the size of an image generator diaphragm. An error in representation of acute angles and other different elements of the topology is determined by the size of the minimum window of an aperture for the generator. A layout precision must be less than 1 discrete coordinate table (0.125 microns for generator EM-5109 and 0.25 microns for EM-5009A2). Further, the optimum sequence of the rectangles is founded to increase the speed up of the generator (the rectangles are sorted on a rotation, because a rotary mechanism in generator type EM-5009A2 is slow).

Table 1 presents processing results for this algorithm on generator EM-5109 with minimum size 1 micron (two hundred polygons from real projects were processed). To reduce the table we show only partial results, but it does not significantly reduce impact on the accuracy and reliability. The results proved that 58% of previous coverings are correct. As a rule, in orthogonal polygons number of rectangles in covering approximately 2 times less than the number of vertices in the original polygon. This is due to the fact that in the implemented algorithm with previous-covering, added a special optimization procedure. According to this procedure we build next rectangle only on the base of segment that is not a side of previously found rectangles.

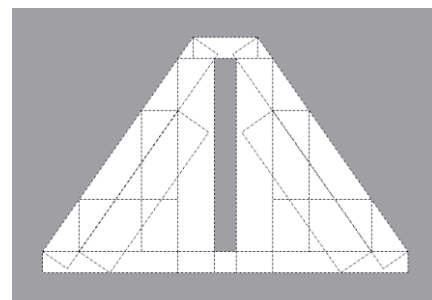


Fig. 7. Covering after second stage

If the input polygon is similar to the bus chip, the resulting covering contains the same number of rectangles as the vertices in the input polygon.

Increasing the number of rectangles in the covering usually associated with the complexity of the original polygons: the presence of circular arcs and sharp corners or defects in bus chip.

5. Conclusions

The proposed algorithm enables to supervise all stages of formation of input sequence for integrated circuit layout generators and process data with usage of technological and technical restrictions of the laser generator. It was imple-

mented as a set of the appropriate subroutines which are integrated in subsystem «Polygon» of the automated system Table 1. Results of data processing

№	The number of vertices in the original polygon	The number of rectangles in previous covering	The number of added holes	The resulting number of rectangles in covering
1	100	42	2	42
2	65	74	0	74
3	97	104	0	104
4	130	146	0	146
5	50	58	0	58
6	66	74	0	74
7	98	104	0	104
8	50	58	0	58
9	100	104	0	104
10	24	24	0	24
11	93	84	0	84
12	127	87	1	88
13	338	286	3	295
14	128	121	6	122
15	20	35	1	38
16	25	39	1	42
17	23	87	2	91
18	25	39	1	42
19	18	37	1	41
20	154	188	0	188
21	240	227	27	296
22	148	154	0	154
23	155	161	0	161
24	148	186	0	186
25	38	39	0	39
26	176	188	14	202
27	166	161	5	169

of input data preparation for IC layout generator. It realizes the multiple document interfaces and, accordingly, consists of a main window, a menu, a set of floating tool bars and affiliated windows opened as necessary. This system essentially facilitates of a designer work. Covering task realized not only in our system but also in CATS FLAT Fracturing

[14], but the proposed algorithm uses a stage of the covering analysis of correctness due to a special algorithm, that guarantees the correctness of covering.

6. References

- [1] Axelrad V. et al: Analysis of microlithography in an open architecture TCAD System. Optical Laser Microlithography IX, Proc. SPIE, 1996, vol. 21-26, p. 648-659.
- [2] Levenson M. D.: Wavefront engineering from 500 nm to 100 nm CD. Optical Microlithography X: Proc. SPIE. 1997. vol. 3051, p. 2-13.
- [3] Iwai H.: Future semiconductor manufacturing-challenges and opportunities. IEDM Teach. Dig., 2004, p. 1-16.
- [4] Feinberg V. Z.: Geometrical problems of machine schedulers for VLSI. M.: Radio and Sviaz, 1987, 178 p.(in Russian).
- [5] De Micheli G.: Synthesis and optimization of digital circuits. New York: McGraw-Hill, 1994, 578 p.
- [6] De Berg Mark, Van Kreveld Marc, Overmars Marc, Schwarzkopf Otfried: Computational Geometry: algorithms and applications. 2nd edition. Springer-Verlag, 2000, 367 p.
- [7] Du Q., Wang D.: Resent progress in robust and quality Delaunay mesh generation. J. of Comp. and App. Math., 2006, v. 195, p. 8-23.
- [8] Berman P., Dasgupta B.: Approximating Rectilinear Polygon Cover Problems. Algorithmica, 17(4), 1997, p. 331-356.
- [9] Kwok-Kit Wong A.: Optical Imaging in Projection Microlithography. Ser. Tutorial Text in Optical Engineering. Bel-lingham, WA: SPIE, 2001.
- [10] Owa S. at al: Current status and future prospect of immersion lithography. Proc. SPIE, 2006, vol. 6154, 61 5408-1.
- [11] Mashohor S., Evans J. R., Arslan T.: Elitist selection schemes for genetic algorithm based printed circuit board inspection system. Evolutionary Comp., 2005, 2, p. 974-978.
- [12] Rubin S.M. Computer Aids for VLSI Design [Electronic resource] - Mode of access: <http://www.rulabinsky.com/cavd/text/chapc.html>. - Date of access: 19.08.2015.
- [13] Technical description and instructions for operating GI EM-5109. Minsk: Planar, 2001, 323 p.
- [14] CATS [Electronic resource] - Mode of access: https://www.synopsys.com/Tools/Manufacturing/MaskSynthesis/CATS/Documents/cats_ds.pdf. - Date of access: 19.08.2015.