Implementation and Testing Algorithm of Input Data Preparation for Single-Beam IC Layout Generator

Avakaw S. [1]**, Doudkin A.**[2]**, Voronov A.**[2]

1) KBTEM-OMO, Partizansky Ave. 2, Minsk, Belarus, office@kbtem.avilink.net, tlo@list.ru

2) United Institute of Informatics Problems of National Academy of Science of Belarus, Surganov str. 6, Minsk, Belarus, doudkin@newman.bas-net.by, voronov@newman.bas-net.by

Abstract: *A new algorithm and a software application for an automated system of input data preparation for integrated circuit layout generator are 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 computer geometry and image analysis.*

Keywords**:** Covering algorithm, VLSI chip, polygon, rectangle, layout generator.

1. INTRODUCTION

There is a problem of formation of layout structures on metallized photo masks by manufacture of integrated circuits, photo-electric converters, LC-indicators, and also many other microelectronic devices [1]. These structures are formed with the help of layout generators. The layout generator builds an integrated circuit (IC) layout on a photo mask from type-setting elements. 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. An input information for the generator is formed as the set of rectangles. There are a lot of the automated systems [1, 2, 4-15] of input data preparation for IC layout generators; however they are focused on a limited class of layout objects. For example CATS is a highly scalable and flexible software application that transcribes complex design data into machine readable instructions for e-beam and laser machines used for a pattern generation and manufacturing of IC, MEMS, TFT-LCD, TFH, photonics, and biochip products. CATS has installations in virtually every photomask manufacturing facility worldwide, and is the de facto standard for mask manufacturing, inspection, metrology, and direct-write-on-wafer. But new versions of CATS software do not support an input data format for equipment EM-5109 and EM-5009A2.

A new algorithm and a software system for an automated system of input data preparation for IC layout generator are proposed in this paper. The system realizes multiple document interfaces and uses step-by-step output mode on the screen elements of the found covering. Also there are some specific capabilities in the system that makes a development and an implementation of algorithms for covering more easily. The software system enables to work with a layout description in GDS II format.

2. LAYOUT DECOMPOSITION

Different tasks are solved during the input data preparation for IC layout generators, but the main of these ones is layout decomposition or covering layout by rectangles (Fig.1).

The decomposition (covering) task is solved for each contour of input data, i.e. a performance of each of these contours as a set of the rectangles is found. Thus each rectangle should satisfy to the given restriction on the minimal length of its side. Special and universal algorithms are used for the decision of a problem of the given stage. Special algorithms solve the covering task of circles, rings, bus-bars and triangles. The universal algorithm solves the covering task of any polygon given by one or several contours. The proposed algorithm is heuristic. It does not guarantee that the received subset of rectangles covers the given contour. Therefore after a reception of the set of the rectangles for covering the task of correctness analysis of this set is solved. For this purpose the rectangles are united in one simply connected contour or a multiply connected contour. If this generated union of the rectangles is the simply connected contour the received covering is correct turns out. If the resulting contour is multiply connected the covering is not correct. Non correct covering is supplemented with the rectangles so that it becomes correct. The obtained sets of rectangles are added in the covering. The prepared input data for IC layout generator you can see on Fig.2. Covering task realized not only in our system but also in CATS FLAT Fracturing [4], but the proposed algorithm uses a stage of the covering analysis of correctness due to a special algorithm, that guarantees the correctness of covering.

Further the optimum sequence of the rectangles is found on the obtained covering set. This sequence is coded using a protocol of an input language for the appropriate IC layout generator. Such task is solved also by software application PG Output Format [4], but a new version of this application does not support GI EM-5109 and GI EM-5009 A2, and our application optimizes input data for such generators.

Thus a number of the rectangles that are enclosed into the result set of rectangles should be minimal or close to minimal, and each rectangle in this set should satisfy to limitations on its sizes. A number of intersections between the rectangles might also be minimized as well as a time for finding covering [8].

The main goal of the software system is to obtain information about input layout patterns in an image format of the generator. The option «Conv» is used to convert graphic images from internal text format into formats MUL, BIT, or PAT, which are an input for layout laser generators EM-5109 and EM-5009A2.

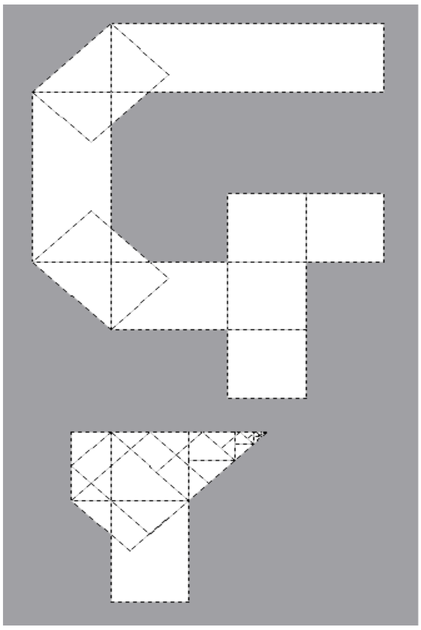


Fig.1 – A result of visualization of output data after covering

One by one we put a description of each rectangle and if necessary cover the rectangles by the elements corresponding to the size of an image generator diaphragm (if GI EM-5109, the sizes are from 1 to 300 microns, if GI EM-5009 A2 - from 2 to 300 microns). It is also necessary to determine the type of generator for which an information is being prepared (EM-5009 A2 or EM-5109), since the accuracy of the final covering for layout patterns is dependent from the corresponding parameters of image generators. Thus, 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 image 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).

To increase the speed up of photomasks production for the generator, if required, all the elements are sorted on a rotation, because a rotary mechanism in generator type EM-5009A2 is slow.

3. NEW ALGORITHM AND IMPLEMENTATION

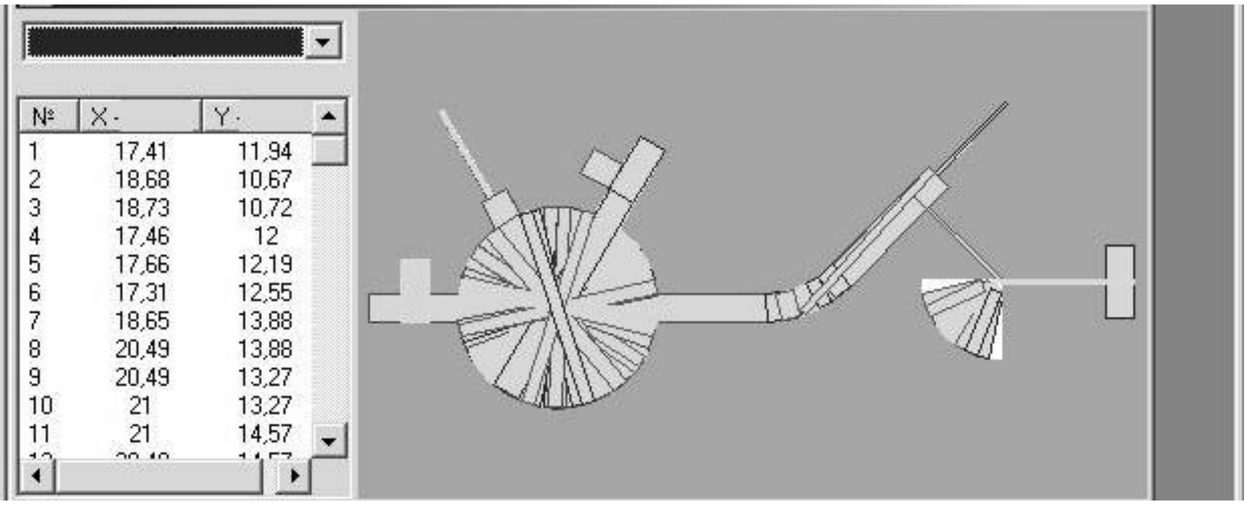
The task described above realized as a set of the appropriate subroutines which are integrated in specially developed visualization system «Polygon». This system essentially facilitates of a designer work. It allows executing complex (transparent) layout designing and the analysis of intermediate results. System Polygon 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.

Different kind of algorithms is realized in the system but we describe covering algorithm with analysis stage.

The covering algorithm consist of two stages: find previous covering, analysis this covering on correctness and find holes in covering. Then special algorithm covers uncovered holes and finds correct covering or entrance sequence for laser generators. Describe in detail task and solution for this algorithm and give some definitions.

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 is a flat shape consisting of straight lines that are joined to form a closed chain or circuit. A *polygon* is traditionally 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, ..., L_g. L_1 is called a main or ground closed polygonal chain and L_2, ..., L_g closed polygonal chains inside main one L_2, ..., 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



№	X ·	Y ·
1	17,41	11,94
2	18,68	10,67
3	18,73	10,72
4	17,46	12
5	17,66	12,19
6	17,31	12,55
7	18,65	13,88
8	20,49	13,88
9	20,49	13,27
10	21	13,27
11	21	14,57

Fig.2 – Part of system «Polygon»operating window with results of visualization prepared input data for integrated circuit layout generators

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 we consider the following problem: we must found for multi connected correct covering polygon W that consists of a minimum or close to the minimum number of h-allowable rectangles.

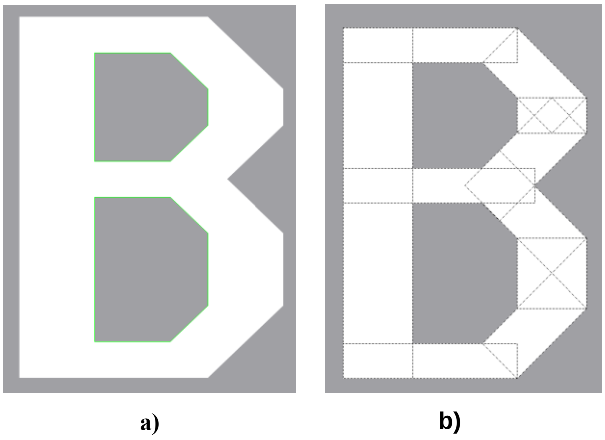


Fig. 3 – Covering a) input polygon; b) covering for input polygon

Example 1. Consider a multiply connected polygon W, shown in Fig. 3. This polygon is given by the main closed polygonal chain L_1 and two contour-cuts: L_2, L_3. The main closed polygonal chain L_1 is the following sequence of vertices, defined by its coordinates (2, 2), (2, 12), (7, 12), (9, 10), (9, 9), (7.5, 7.5), (9, 6), (9, 4), (7, 2). Contour-cuts L_2, L_3 respectively, are defined by sequences of vertices: (4, 8), (4, 11), (6, 11), (7, 10), (7, 9), (6, 8);(4, 3), (4, 7), (6, 7), (7, 6), (7, 4), (6, 3).

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

Below for simplicity polygon sometimes will be denoted by Latin letters a, b, c, etc. Denote by $P(ab)$ 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 is not an 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. 4). Among the points of the polygon, that lie between these two lines and the side a, we looking 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.

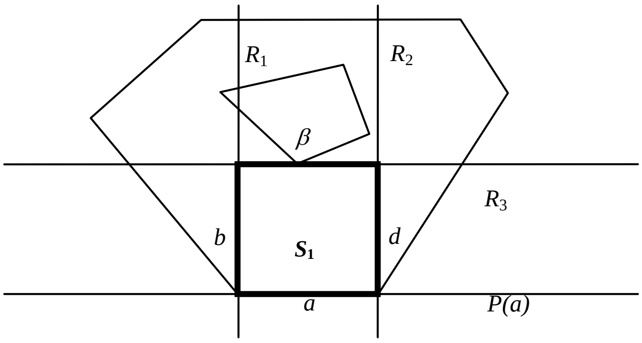


Fig. 4 – Previous covering by main rectangle

Further we extend the main rectangle S_1 to its sides d and b. The basic or main rectangle shown on Fig. 4 cannot be extended to these sides. Therefore, we turn to Fig. 5.

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 (Fig. 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.

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. 6). 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, adjacent to polygon side a and connecting with it in polygon vertex a_1 a1so intersects with line F_1.

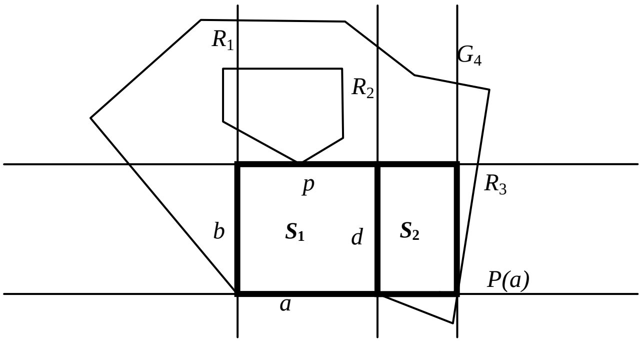


Fig. 5 – Previous covering after extension main rectangle by side *d*

Denote as T line on with polygon side t located, and A line on with polygon side a located. Denote F_2 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 parallel to line A and α also must situated on this line. Find point β, this is the point of intersection lines F_3 and T. Draw the line through this point F_4, which parallel to line F_1. Among the points of multiply connected polygon, that lie between lines F_1, F_4 and A, find point γ, that located on the boundary of polygon and on minimum distance from side a.

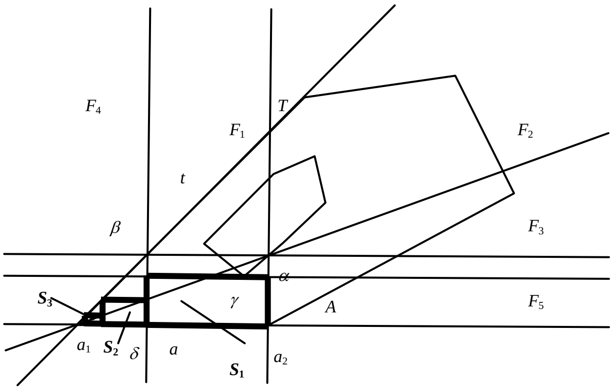


Fig. 6 – Construction rectangles for polygon side *a*, one of polygon interior angles on side *a* acute (variant 1)

Through this point draw straight line F_5 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 located at the intersection of lines F_1, F_4, F_5, A set end points for rectangle S_1 (Fig. 6), else points located at the intersection of lines F_1, F_4, F_3, A set end points for rectangle S_1. 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 the side a put under consideration line segment, 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_1δ of the side a stops because we build only h-allowable rectangle. Otherwise, we add rectangle S_2 in the previous covering, and similarly construct rectangle S_3 (Fig. 6), 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. 7). 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, α), (α, 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).

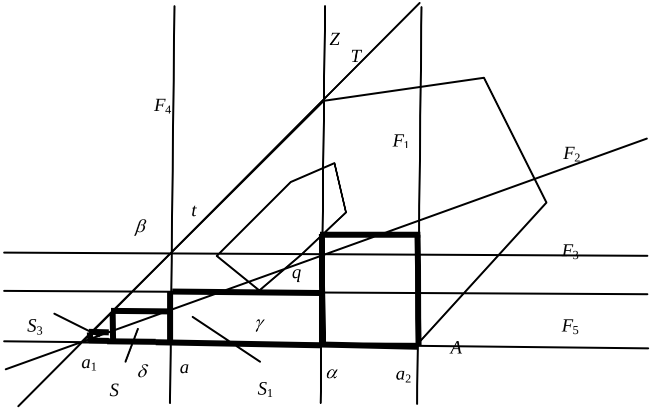


Fig. 7 – Construction rectangles for polygon side *a*, one of polygon interior angles on side *a* acute (variant 2)

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. 7), can be constructed several rectangles for previous covering.

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), (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. In next Example 2 previous covering is incorrect.

Example 2. On Fig. 8 shows a multiply connected polygon and on Fig. 9 its previous covering obtained by the algorithm described above. Obtained covering is incorrect, since it contains two non-covered holes.

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 previous covering on correctness and cover added at first stage contour-cuts. On Fig.4 two contour-cuts or holes added arter first stage.

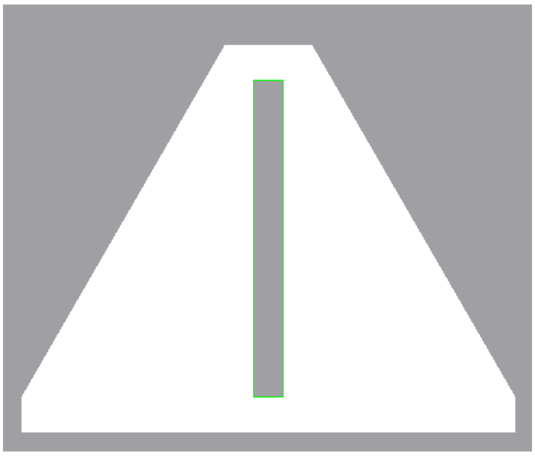


Fig. 8 – Polygon before covering

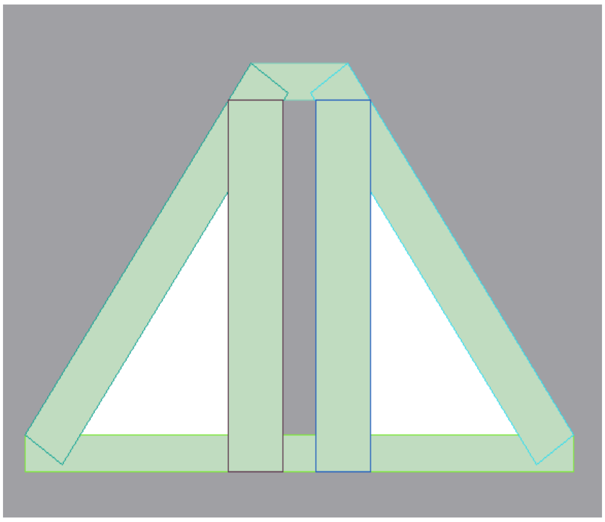


Fig. 9 – Covering after first stage

After such analysis we cover contour-cuts that we find, Fig. 10.

This algorithm implemented in system and display good results for example in Table 1 you can see processing result for this algorithm on generator EM-5109 with minimum size 1 micron Table 1 represents results of data processing two hundred polygons from real projects. To reduce the table we show the every five result, but the table does not significantly reduce impact on the accuracy and reliability.

So for 58 % of the polygons previous covering proved 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.

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.

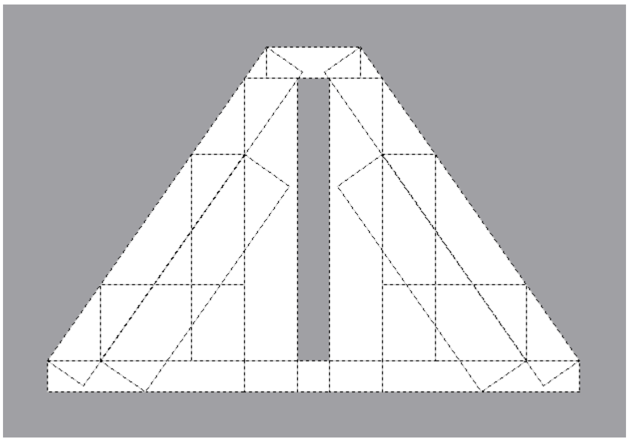


Fig. 10 – Covering after second stage

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	10	4	0	4
12	12	7	0	7
13	16	37	0	37
14	16	10	0	10
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	20	34	1	35
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	9	46	6	52
28	166	161	5	169
29	16	37	0	37
30	16	33	0	33
31	16	37	0	37
32	23	88	2	90
33	14	34	1	38
34	47	62	2	67
35	9	15	0	15
36	9	16	0	16
37	9	42	0	42
38	168	242	34	287
39	129	216	31	256
40	139	157	4	161
41	164	130	2	132
42	338	286	3	295
43	128	121	6	122
44	240	227	27	296
45	5	13	0	13

46	259	173	8	187
47	183	108	0	108
48	63	49	2	55
49	182	107	0	107
50	93	84	0	84
51	127	87	1	88

4. CONCLUSION

The proposed software 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. There are alternative variants of decision decomposition task and as you can see in Table 1 some algorithms implemented and display good result. All this enables to form better entrance sequences for integrated circuit layout generators.

5. REFERENCES

[1] Axelrad, V. Analysis of microlithography in an open architecture TCAD System. / Axelrad, V. [et al] // Optical Laser Microlithography IX, Proc. SPIE, Vol. 21–26, 1996 P. 648–659.

[2] Technical description and instructions for operating GI EM-5109. – Minsk, “Planar”, 2001. – 323 p.

[3] GDS II format. / http://www.xs4all.nl/~kholwerd/ interface/bnf/gdsformat.html

[4] CATS. / http://alt-s.ru/catalog/synopsys/dfm/cats. html

[5] Levenson, M. D. Wavefront engineering from 500 nm to 100 nm CD. / M. D. Levenson // Optical Microlithography X: Proc. SPIE. – 1997. – Vol. 3051. – P. 2-13.

[6] Iwai, H. Future semiconductor manufacturing-challenges and opportunities / H. Iwai // IEDM Teach.Dig., 2004, P. 1–16

[7] Feinberg, V.Z. Geometrical problems of machine schedulers for great IC // M.: Radio and Sviaz 1987. – 178 p.(in Russian)

[8] De Micheli, G. Synthesis and optimization of digital circuits / G. De Micheli. – New York: McGraw-Hill, 1994. – 578 p.

[9] De Berg Mark, Van Kreveld Marc, Overmars Marc, Schwarzkopf Otfried / Computational Geometry: algorithms and applications. 2nd edition. Springer-Verlag, 2000, 367 p.

[10] Chin, F. Finding the Medial Axis of a Simple Polygon in Linear-Time / F. Chin, J. Snoeyink, C.A. Wang // ISAAC '95, Cairns, Australia, 1995 (LNCS 1006 Springer-Verlag).

[11] Du, Q. Resent progress in robust and quality Delanay mesh generation. / Q. Du, D. Wang // J. of Comp. and App. Math., 2006, V.195, P. 8–23.

[12] Berman, P. Approximating Rectilinear Polygon Cover Problems / P. Berman, B. Dasgupta // Algorithmica, 17(4), 1997, P. 331–356.

[13] Wong, A. Kwok-Kit Optical Imaging in Projection Microlithography / A. Kwok-Kit Wong // ser. Tutorial Text in Optical Engineering. Bellingham, WA: SPIE, 2001.

[14] Owa, S. Current status and future prospect of immersion lithography / S. Owa [et al] // Proc. SPIE, 2006, vol.6154, 61 5408–1.

[15] Matjuskov, V.E. Laser process equipment for precission treatment of materials in electronics / V.E. Matjuskov, S.M. Avakaw // Proceedings of Laser Technology: Indo-Belarus Workshop, New Delhi, India, 2003