PLACEMENT OF VLSI CIRCUITS’ FRAGMENTS BY THE PROCEDURE OF DICHOTOMOUS DIVISION

Aleksandr Nazarov, Konstantin Smirnov

Abstract. The purpose of the present work is to minimize time for and improve the quality of automatic placement of fragments of Very Large Scale Integration circuits on a crystal as compared to the known algorithms. Modern methods of solving this problem use search procedures, which produce the results appropriate for use in practice, requiring much computer time. Method of placement, suggested in this paper, instead of search procedures uses a procedure of dichotomous division of crystal area into sectors with further redistribution of VLSI fragments across crystal. Sector-wise placement of VLSI fragments implements a formal mechanism of “attracting” fragments to each other that optimizes the structure of connections between fragments. The method allows to optimally place several thousands of VLSI fragments on a crystal in minutes. Placement enables to reduce the total length of connections between fragments and order their structure. The paper presents the results of comparing effectiveness of the suggested algorithm with effectiveness of the method for placing VLSI fragments using fractal aggregation and genetic algorithm. A gain (≈12.5%) is yielded in total length of connections, with substantially shorter time duration for processor functioning.

Key words: very large scale integration circuit, automatic placement, NP-complete problem, procedure of dichotomous division, attraction criterion, algorithm of fractal aggregation.

Introduction

Nowadays, while developing electronic equipment, more than 65% of time is spent on designing very large scale integration circuits [2–4, 6]. Here, the most challenging of all problems related to technical design of VLSI circuits is a problem of optimal placement of their fragments. Despite multitude of attempts to address this problem, it remains problematic and relevant to the present day.

Work [7] presents this problem as “quadratic problem of assignments”. It is impossible to find its accurate solution even at number of VLSI fragments that exceeds 10. Indeed, searching for its optimal solution with an adequate global minimum of selected placement criterion, results in the need for implementing all n! permutations of fragments on regular lattice. Such problems are NP-complete, and, therefore, all the known methods of its solving [9, 12, 16, 17] produce only approximate results. The group of sequential so called “fast” algorithms shall be immediately excluded from analysis, since their low effectiveness is proved [14].

The group of search or hybrid algorithms [19, 20] appears definable among methods [12] for its solution. These algorithms involve the most efficient algorithm for placing VLSI fragments on the basis of fractal aggregation [5] and genetic algorithm [18, 21]. As is true with all search methods, this algorithm requires substantial processor operation time, however, it allows to obtain the result, quite relevant for use in practice.

As is known, the main purpose of placing VLSI fragments is to create the most favorable conditions for further trace of interconnects on crystal [13]. Thus, it makes sense to control the structure of connections in the course of optimizing placement [15]. Unfortunately, literature on automated engineering design of electronic equipment gives little attention to this problem, and to its formal aspect, in particular. Only worded descriptions of common approach to solving this problem are known [7, 8]. The key phrases listed here involve “compact placement”, “tightly coupled groups of elements”, and other. They do not enable to practically use the description, presented in the mentioned works, or imply only its manual use. More thorough analysis of the above sources allows to summarize the following practical finding, which were taken as a basis when developing the formal placement algorithm suggested in the paper. Algorithm shall implement an iterative procedure of integrating VLSI fragments on a crystal into tightly coupled groups (sectors) in the way that fragments from the other similar groups are prohibited from being inside each of them, even in case if these similar groups have vacant positions for placing. It is a principle of dichotomous division (into halves) that is perfectly suitable to formalize the above.
Suggested method

The algorithm uses a bipartite graph \( G(X, Z, R) \) as initial data, in which a multitude of the first portion of vertexes \( X = \{x_1, x_2, \ldots, x_n\} \) models VLSI fragments placed on a crystal; multitude of the second graph portion of vertexes \( Z = \{z_1, z_2, \ldots, z_m\} \) models equipotential nodes of electric diagram, hereinafter called circuits, and multitude \( R \) is a multitude of binary edges of type \( x_i \rightarrow z_j \), that model all proper electric circuits of the initial electric diagram.

Fig. 1 shows the example, clarifying the definitions introduced, where:

\[
X = \{x_1, x_2, x_3\}, \quad Z = \{z_1, z_2, z_3\}, \quad R = \{x_1 \rightarrow z_1, \, x_2 \rightarrow z_4, \, x_3 \rightarrow z_1, \, x_4 \rightarrow z_2, \, x_5 \rightarrow z_3, \, x_1 \rightarrow z_2, \, x_2 \rightarrow z_2\}
\]

![Example of bipartite graph, modelling 3 fragments of VLSI](image)

Fig. 1. Example of bipartite graph, modelling 3 fragments of VLSI

The essence of algorithm is to iteratively apply the following four procedures:

- all the field of positions to place VLSI fragments is divided into two sectors, which are filled with tightly coupled groups of fragments;
- each of the two new sectors shall be once more divided into two sectors, and fragments shall be distributed across these new sectors;
- the next \( i \)th iteration of dividing sectors shall form new sectors, in each of which the number of positions is twice less;
- at step \((n-1)\) division terminates, since in each of \( n \) sectors there is placement position.

It is advisable to commence implementation of the first two steps of dichotomous division from constructing a discrete working area, which will be assumed to be a rectangular lattice of \( N \times M \) discretes’ dimension. It is allowed to place VLSI fragments in the nodes of this lattice. The only restriction shall be imposed on the number of horizontal \( N \) and vertical \( M \) rows of lattice: each of them shall be \( 2^k \)-fold, where \( k = (1, \, 2, \, \ldots) \). The total number of steps of iterative (dichotomous) division here will amount to \((M \times N - 1)\). Thus, for discrete working area of \( M = N = 64 = 2^6 \) side only \( 64 \times 64 - 1 = 4095 \) steps will be required. And, if it is considered that each placement position may contain one fragment, maximum number of fragments to be placed will be equal to \((n = 4096)\).

For computer-assisted implementation of thereby introduced discrete working area, real numbers \( S \) of vertical and horizontal rows of placement positions shall be added with numbers \( P \) and \( Q \) of positions, respectively. Then, \( N = S + P \) and \( M = T + Q \) shall be chosen as equal to a multiple of integral power of two. For example, if \( S = 3 \), then \( P = 1 \); if \( T = 6 \), then \( Q = 2 \) and so on.

Fig. 2 presents an illustration of the procedure for adding horizontal and vertical rows in a discrete \( 5 \times 3 \) working area (here, the added, forbidden to be placed positions are highlighted in grey). The numbers near cut lines show the order of dichotomous division of the discrete working area into sectors.
It can be strictly shown that the described approach to constructing a discrete working area will ensure maximum speed of dividing a crystal into sectors. An algorithm of implementing the stage of distributing fragments across sectors, requires introduction of a criterion for attracting fragments to each other within the same chain, and to fragments of the other chains. A rational choice of the indicated criterion will allow to control the procedure for moving VLSI fragments within the sector, selected for division, in a focused manner. This procedure shall be implemented in such a way as to ensure the resulting integration of all fragments into tightly coupled groups, i.e. this procedure should algorithmically forbid presence of fragments from the other tightly coupled groups inside the \(i^{th}\) group, ignoring the fact that there are vacant positions for their placement inside the \(i^{th}\) group.

A quality aspect of the problem shall first be considered through drawing attention to Fig. 3.

Two trees of bipartite graphs are shown in these figures, which model two equipotential chains \(z_1\) and \(z_2\) of a principal diagram, connecting fragments of VLSI, represented by vertexes of the second portion of graph: \(A, B, C, D\) and \(E\). Cut line \(S - S\) divides these vertexes into two groups, and the crossed position is vacant. It is required to move one of elements in the right sector from the right to the left and place it to the specified position. If element \(B\) is chosen, it is easier to implement connection \(AB\), and the number of inter-sector connections decreases, as it is shown in Fig. 3.b. So as element \(B\) is also “chosen” by a computer, such criterion of “attraction” shall be introduced for it, which would be greater than the similar criterion for elements \(D\) and \(E\).

The same-name “attraction criterion”, hereinafter called “attraction” and designated as \(t_j(i)\), shall be introduced as such a criterion of attraction of the \(i^{th}\) vertex to the other vertexes of the \(j^{th}\) chain. To formally describe it is most easily done by using the language of graphs’ theory. To this end, an universal quantifier: \(\Gamma v\) shall be taken into consideration, where

\[ \Gamma v \]

is a random vertex of bipartite graph \(G(X, Z, R)\). By definition, universal quantifier \(\Gamma v\) integrates a multitude of vertexes, incident to vertex \(v_i\) of graph \(G(X, Z, R)\). For example, for the graph, depicted in Fig. 2 the following
expressions are true: \( \Gamma z_j = X_1 = \{x_1, x_2, x_3\}, \Gamma x_1 = Z_1 = \{z_1, z_2, z_3\}, \Gamma z_2 = X_1 = \{x_1, x_2\} \) and so on. The introduced designations will enable to determine attraction of \( i - \)th vertex to vertexes of \( j - \)th chain as:

\[
t_{z_j} [\forall x \in \Gamma z_j] = \frac{1}{|\Gamma z_j|}
\]

Thus, computation of criterion for attraction of each vertex from multitude \( \Gamma z_j \) by formula (1) produces the following results:

\[
t_{z_1} [x_1] = t_{z_1} [x_2] = t_{z_1} [x_3] = \frac{1}{|\{x_1, x_2, x_3\}|} = \frac{1}{3} \approx 0,33
\]

An initial placement of vertexes in the positions shown in Fig.3,a shall be considered. Cutline \( S - S \) divides them into two sectors. Suppose that it is required to choose one of the three alternative vertexes \( \{x_1, x_2, x_3\} \) of the right sector to transfer to the position marked with a cross. The purpose of moving is to decrease the number of connections between sectors. The introduced attraction criterion shall be used to this end. Calculations by formula (1) produce the following result:

\[
t_{z_1} [x_1] = \frac{1}{|\{x_1, x_4\}|} = 0,5
\]

\[
t_{z_1} [x_4] = \frac{1}{|\{x_1, x_4\}|} = 0,5
\]

\[
t_{z_1} [x_2] = \frac{1}{|\{x_5, x_2, x_3\}|} \approx 0,33
\]

\[
t_{z_2} [x_3] = \frac{1}{|\{x_5, x_2, x_3\}|} \approx 0,33
\]

\[
t_{z_2} [x_3] = \frac{1}{|\{x_5, x_2, x_3\}|} \approx 0,33
\]

Through comparison of the obtained attraction criteria, a choice shall be made of vertex \( X_1 \) with the maximum attraction, and it is observed that a “manual” selection by method of simple enumeration shall produce the same result.

In Fig.4 two vertexes are located in the sector of choosing \( C0 \), and it is required to place one of them in sector \( C1 \), and the other – in sector \( C2 \). Since in this case there are two directions of moving vertexes horizontally, and, in general case, there are two more vertical directions, it is wise to identify each of them. Let those vertexes that tend to increase the coordinates of vertexes related to them, have sign “plus” to this end. The similar situation takes place for vertexes \( X_3 \) and \( X_4 \), which tend to increase the coordinates of vertex vertexes \( X_5 \) and \( X_6 \), respectively, attracting them by their chains.

![Fig.4](image_url)

**Fig.4.** Two vertexes in sector of choosing
Similarly, sign “minus” shall be assigned to vertexes that tend to decrease the coordinates of vertexes related to them. Similar situation takes place for vertexes \( X_1 \) and \( X_2 \), which tend to decrease the coordinates of vertexes \( X_5 \) and \( X_6 \). Finally, when calculating criterion \( t_j(i) \) for \( i \)-th vertex, a sign for attraction shall be chosen equal to a sign of vertex \( i \). Considering the above, attractions computed by formulas (2), (4) and (5) will have signs “plus”, respectively, and by formulas (3) and (6) - signs “minus”.

The introduced rule of signs allows calculating the total attraction \( t_{\Sigma} (x_i) \) for vertexes \( X_5 \) and \( X_6 \) of sector C0 in Fig.4. To this effect, it is sufficient to find a sum of attractions for each vertex across all chains, where they are available. Below are the results of these calculations:

\[
\begin{align*}
\quad t_{\Sigma} (x_5) &= t_{z_1} (x_5) + t_{z_4} (x_5) = (-0,33 - 0,33) + (0,25 + 0,25) = -0,16 \\
\quad t_{\Sigma} (x_6) &= t_{z_2} (x_6) + t_{z_1} (x_6) + t_{z_4} (x_6) = (-0,5) + (-0,33 + 0,33) + (0,25 + 0,25) = 0
\end{align*}
\]  

(7)  

(8)

Comparing both obtained attraction criteria, a solution shall be adopted for moving vertex \( X_5 \) to the left sector and vertex \( X_6 \) - to the right one. The result is given in Fig.4 to theright.

A rectangular table is a comfortable computational model for calculating the total attraction of sector C0 vertices, which will hereinafter be called a matrix of attractions \((MA)\). By definition, element of matrix \((MA)\), located at the intersection of \( i\)-th line and \( j\)-th column is equal to \( t_{j(i)} \), where:

- \( i \)-index of vertex of multitude \( X \subseteq G(X, Z, R) \),
- \( j \)-index of vertex of multitude \( Z \subseteq G(X, Z, R) \).

For the current example, matrix \((MA)\) is given in Table 1. Here in column \( M \) sign “plus” is written opposite to vertexes of the right sector, and sign “minus” is opposite to vertexes of the left sector. Further, the marked vertexes are called poles. Vertexes, which fall into division sector at the current step of division, do not have signs. Below is the intermediate algorithm of calculating total attraction for particular vertex \( X_5 \) of this sector:

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( z_j )</th>
<th>( M )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_1 )</td>
<td>( z_1 )</td>
<td>( 0,33 )</td>
</tr>
<tr>
<td>( x_2 )</td>
<td>( 0,33 )</td>
<td>( 0,5 )</td>
</tr>
<tr>
<td>( x_3 )</td>
<td>( 0,25 )</td>
<td>( +)</td>
</tr>
<tr>
<td>( x_4 )</td>
<td>( 0,33 )</td>
<td>( 0,25 )</td>
</tr>
<tr>
<td>( x_5 )</td>
<td>( 0,33 )</td>
<td>( 0,33 )</td>
</tr>
<tr>
<td>( x_6 )</td>
<td>( 0,33 )</td>
<td>( 0,5 )</td>
</tr>
</tbody>
</table>

\( \alpha_1 \) - after having looked through line \( x_5 \) of matrix \((MA)\) prior to meeting its non-zero element (here, it is element 0.33 for chain \( Z_3 \)), the second item shall be examined; if this element is not found, the third item \( \alpha_3 \) shall be analyzed to complete computations of attraction \( t_{\Sigma}(x_5) \);

\( \alpha_2 \) - column \( Z_3 \) shall be looked through, by successively adding to sum \( t_{\Sigma}(x_5) \) those its non-zero elements (attractions), which have a sign in column \( M \). Here, the number from column \( Z_3 \) shall be taken considering this sign: for the current example after looking through column \( Z_3 \) the following portion of a required sum shall be obtained:
\[ t_\Sigma (x_5) = -0,33 - 0,33 + ...; \]

After having looked through column \( z_3 \), item \( \alpha_1 \) shall be analyzed to continue looking through line \( x_5 \):

\[ \alpha_3 \] – at the moment of transferring control to this step, the same calculations will be made as those defined by formula (7).

It may be seen that similar calculation of the total attraction for vertex \( x_6 \) will be made by formula (8). A procedure-function of computation in cell of the total attraction for \( i \)-th vertex shall be called from the principal program by operator \( \text{attract}(i) \).

More general example, given in Fig. 5, shall further be considered of optimizing placement of VLSI fragments. Here, it is required to improve the initial placement of fragments marked with numbers 9-12 in the field of positions, presented in Fig. 5.a. Fragments from 1 to 8 are fixed. Algorithm provides for implementing the following stages:

1. Compiling matrix of attraction criteria (MA), given in Table 2.

2. Dividing central four positions, highlighted with colour in Fig. 5.a by vertical cut line \( A - A \). Since fragments 1-4 are found to the left from cut line \( A - A \) and beyond boundaries of the central zone, they take sign “minus” in column \( M_1 \) of attraction matrix. Similarly, elements 5-8, found to the right from the cut line, will have sign “plus”.

![Fig. 5 Initial (a) and resulting (b) placement of vertexes](image)

**Table 2**

<table>
<thead>
<tr>
<th>( i )</th>
<th>( j )</th>
<th>( M_1 )</th>
<th>( M_2 )</th>
<th>( M_3 )</th>
<th>( M_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0.25</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>0.25</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>0.25</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>0.25</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>0.25</td>
<td>0.33</td>
<td>0.33</td>
<td>+</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>0.33</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>0.33</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>0.33</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td>0.33</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>+</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>0.33</td>
<td>0.33</td>
<td>0.33</td>
<td>-</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>12</td>
<td>0.25</td>
<td>0.25</td>
<td>0.25</td>
<td>-</td>
</tr>
</tbody>
</table>

Then, actions are taken according to the above algorithm of computing total attraction. For the current example, attraction shall be calculated for the elements, not marked in column \( M_1 \):

\[ t_\Sigma (9) = +0,33 + 0,33 + 0,33 = +1; \]
$t_e(10) = +0,33 + 0,33 + 0,33 + 0,33 = +1,33$;  
$t_e(11) = -0,25 + 0,25 + 0,33 - 0,33 + 0,25 + 0,25 - 0,25 - 0,25 = -0,92$;  
$t_e(12) = 0,25 - 0,25 - 0,25 + 0,25 = -0,5$.

$10^{th}$ element (11$^{th}$ element) has maximum (minimum) attraction, therefore it is transferred to the right (left) sector from cut line A-A and has sign “plus” (“minus”), which is reflected in column $M_2$ of the matrix of attractions. Signs for elements 1-8 remained the same. New attractions shall be calculated for the rest unmarked elements 9-12. For vertex 9:  
$t_e(9) = +0,33 + 0,33 + 0,33 - 0,33 + 0,33 + 0,33 = +1,33$,  
and vertex 12:  
$t_e(12) = -0,25 - 0,25 - 0,25 + 0,25 - 0,25 = -1$.

The first step of division shall be finished by transferring element 9 and 12, respectively, to the right and left sectors (Fig. 5, b).

3. The left half of the central sector shall be divided by horizontal cut line B-B. Since elements with 1, 2, 5, and 6 are found to be above cut line B-B, and beyond the central zone, then they have sign “plus” in column $M_3$. Similarly, elements 3, 4, 7, and 8, below the cut line obtain sign “minus”.

The total attraction shall be computed for the elements not marked in column $M_3$, which belong to the left part of the “coloured” sector:  
$t_e(11) = +0,25 + 0,25 - 0,33 + 0,25 + 0,25 - 0,25 - 0,25 = +0,42$;  
$t_e(12) = +0,25 + 0,25 + 0,25 + 0,25 = +1$.

The 12$^{th}$ element with greater attraction shall be transferred to the upper sector, and the 11$^{th}$ element with lower attraction shall be transferred to the lower sector. After correction of signs (column $M_4$), the right half of central sector shall be similarly divided, and the following values of attractions for elements 9 and 10 shall be obtained:  
$t_e(9) = -0,33 - 0,33 - 0,33 + 0,33 = -0,66$;  
$t_e(10) = -0,33 + 0,33 + 0,33 + 0,33 = +0,66$.

The result of placement (Fig. 5, b) compares favorably with the initial one: all intersections of edges, as potential sources of traces’ conflicts when dividing, disappeared, and their length reduced by 30%.

To assess effectiveness of the suggested algorithm, it will be compared with the algorithm of placing VLSI fragments on the basis of fractal aggregation mechanism (FA) implemented in work [5] through joining algorithms and genetic algorithm. Fractal aggregation mechanism at its first step integrates (aggregates) VLSI fragments, located in each horizontal row, into a certain group. As a result, there appears a monopartite graph in the form of vertically located vertexes or “floors” connected by vertical edges, and each vertex of graph obtained in this manner accommodates all those connections that were inherent in its component fragments prior to aggregation. In the second step of algorithm work, a genetic algorithm shall be implemented that finds a quasi-optimum floor-wise permutation of aggregated vertexes. A reverse unpacking of aggregated fragments completes the work of algorithm.

As the authors of the algorithm have shown, the quality of placements obtained on its basis (i.e. on the basis of combined search) is by 6.38% on the average higher than that of placement results obtained using the known algorithms Capo 8.6, Feng Shui 2.0, Dragon 2.23…[1].

Since work [5] presents a particular example of fractal aggregation algorithm work, for purposes of comparing effectiveness, it is advisable to use the same patch board of fragments in the form of graph $G = (X, U)$, located in lattice $3 \times 3$ (Fig. 6, a).

When computing the total length of connections in the test case, Manhattan geometry [11] and a unit step between lattice nodes were applied to calculate distances between vertexes. As a result, the total length of connections of the initial placement (102) was reduced to 96 in the final placement, presented in Fig. 6, b.

The stages of placing this example by the suggested algorithm shall further be considered. Figure 7, a shows a field of placement positions prior to the 1$^{st}$ step of dividing this field by halves with vertical cut line S1. It is prohibited to place fragments in the darkened positions. Vertexes within darkened area is fixed and further used as a pole (hereinafter only indices of vertexes are shown in the patch board).
Fig. 6. Initial (a) and resulting placement of fragments by algorithm for aggregation of fractals (b)

(a)  
```
1 3 2  
4 2 1  
7 8  
```

(b)  
```
1 3 2  
4 2 1  
7 8  
```

Fig. 7. Placements of fragments: initial (a), after the 1st step of dividing (b)

(a)  
```
1 2 3  
4 5 6  
7 8 9  
```

(b)  
```
1 3 4  
6 9 5  
2 8 7  
```

Initial MA matrix is given in Table 3. In compiling it, each edge of a monopartite graph in Fig. 6, a is presented in the form of one chain of a bipartite graph. For example, eight connections between vertexes $X_1$ and $X_3$ represent the first 8 columns of MA matrix, and so on. Table 4 presents its equivalent matrix in short. Column «M» contains sign “minus” opposite to vertex $X_1$, since it attracts related to it vertexes towards a decrease in coordinates.

Table 3

<table>
<thead>
<tr>
<th>Z1 - Z32</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5</td>
</tr>
<tr>
<td>X2 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5</td>
</tr>
<tr>
<td>X3 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5</td>
</tr>
<tr>
<td>X4 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5</td>
</tr>
<tr>
<td>X5 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5</td>
</tr>
<tr>
<td>X6 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5</td>
</tr>
<tr>
<td>X7 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5</td>
</tr>
<tr>
<td>X8 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5, 0,5</td>
</tr>
</tbody>
</table>

The remaining cells of column «$M_1$» contain values of the resultant attractions of vertexes $X_2 + X_8$ to vertex $X_1$. Vertex $X_3$ has a maximum attraction to vertex $X_1$, therefore it is transferred to the left sector and marked with sign “minus” in column «$M$». The described procedure is repeated 4 times more until positions of the left sector are filled. The results of calculations are given in columns $M_2 - M_6$. According to the results of computations, vertexes with maximum values of attractions, marked with quadrates in the figure, shall be entered to the left sector as follows:

$$X_3 \rightarrow X_6 \rightarrow X_9 \rightarrow X_2 \rightarrow X_8$$

The remaining vertexes $(X_4, X_5, X_7)$ are randomly placed to the right sector (figure 7, b).

Table 4
Figures 8 and 9 present the results of computing criteria of attraction and redistribution of vertexes, when performing dichotomous bi-division of placement field by cut-lines S2–S7. Forbidden positions are not shown for brevity.

The resultant placement of fragments is shown in Fig.10. Calculation of the total length of connections of the resultant placement, using the same methodology, as for the test case, produces the following result: 84p.u., i.e. by 12.5% better than the result of algorithm for aggregating fractals.
number of placed fragments of the project in the first step is \( n = n_1 \);
number of placed chains of the project is \( M \);
model of the diagram for connecting VLSI fragments has a form of a bipartite graph;
two-dimensional matrix of attractions \( MT_{n \times m} \) and linear array \( U \) with coordinates of placement sectors are previously formed;
all vertexes of multitude \( X \subseteq G(X, Z, R) \) of graph are randomly placed in the nodes of initial discrete working area.

Algorithm implies implementation of the following steps:

1. Add to \( N \) vertical and \( M \) horizontal rows of placement positions of discrete working area \( P \) vertical and \( Q \) horizontal rows so as inequalities: \( (N + P) = (M + Q) = n = 2^k \) are met. The added position shall be prohibited for placement of vertexes thereupon from multitude \( X \subseteq G(X, Z, R) \). Value 1 shall be assigned to sector counter \( j \).

2. Choose coordinates of the left bottom and right top angles of the \( j^\text{th} \) sector of division into cells:
   \[
   X_1 \leftarrow U(j), \quad Y_1 \leftarrow U(j + 1), \quad X_2 \leftarrow U(j + 2), \quad Y_2 \leftarrow U(j + 3)
   \]

3. If sector is quadratic (i.e. \( X_2 - X_1 = Y_2 - Y_1 \)), to specify a direction key \( \text{Key}=0 \) and proceed with item 4, otherwise, to specify a direction key \( \text{Key}=1 \) and compute an ordinate of the \( j^\text{th} \) cut line by formula: \( Y_j = \left\lceil \frac{Y_2 - Y_1}{2} \right\rceil + 1 \).

Then, to assign to each \( k^\text{th} \) vertex out of circuit, restricting the divided sector, a sign “minus”, if its ordinate is \( Y_k \leq Y_j \) and sign “plus”, if otherwise, and proceed with item 5 of the algorithm.

4. To compute abscissa of the \( j^\text{th} \) cut line by formula: \( X_j = \left\lceil \frac{X_2 - X_1}{2} \right\rceil + 1 \) and assign to each \( k^\text{th} \) vertex out of circuit, restricting the divided sector, sign “minus”, if its abscissa is \( X_k \leq X_j \) and sign “minus”, if otherwise.

5. Specify a pair of vertexes with extremum attractions, using the above-introduced program.

If \( \text{Key}=0 \), to specify a vertex with a minimum (maximum) attraction to the left (right) of cut line, and assign a sign “minus” (“plus”) to it. Otherwise, to specify a vertex with a minimum (maximum) attraction below (above) cut line and assign a sign “minus” (“plus”) to it.

6. Repeat the previous item \( \left\lfloor \frac{n_j}{2} \right\rfloor \) times, modify counter \( j \leftarrow j+1 \) and proceed to item 2, if \( j < n \), and if \( j = n \), to finish the computations.

A procedure for forming linear array \( U \) mentioned in the algorithm, with coordinates of placement sectors shall be given on original Delphi language for brevity.

Array \( U \) as such shall first be defined:

```delphi
Var U: array[1..8192] of record x1,y1,x2,y2: Real;
```

Then a procedure for its creating shall be as follows:

```delphi
With Reg[1] do begin x1:=1000;y1:=1000;x2:=4200;y2:=3400 end;
  i:=1; j:=2; Count:=1 //начало формирования координат секторов
repeat
  For k:=1 to Count do
  begin DelReg(i,j); i:=i+1; j:=j+2 end;
  Count:=Count*2;
until Count:=4096;
```
This procedure on the basis of the original sector with specified coordinates of its opposite corners, generates in cells $x_1 = y_1 = 1000$ and $x_2 = 4200$, $y_2 = 3400$ a multitude of similar sectors of twice less area at each step. The next pair of new sectors shall be written one by one into array $U$ by index through procedure $\text{DelReg}(i,j)$, which acquires index $i$ of the next sector to be divided, as a parameter. The original text of this procedure has the following form:

```verbatim
Procedure DelReg(Var i,j: integer); var xL,yL,xR,yR,Lx,Ly: real; Begin xL:=Reg[i].x1; xR:= Reg[i].x2; Lx:=(xR-xL) / 2; yL:=Reg[i].y1; yR:= Reg[i].y2; Ly:=(yR-yL) / 2; if Lx>Ly thenWith Reg[i] do begin Reg[j].x1:=x1; Reg[j].x2:=x1+Lx; Reg[j].y1:=y1; Reg[j].y2:=y2; Reg[j+1].x1:=x1+Lx; Reg[j+1].x2:=x2; Reg[j+1].y1:=y1; Reg[j+1].y2:=y2 end; elseWith Reg[i] do begin Reg[j].x1:=x1; Reg[j].x2:=x2; Reg[j].y1:=y1; Reg[j].y2:=y1+Ly; Reg[j+1].x1:=x1; Reg[j+1].x2:=x2; Reg[j+1].y1:=y1+Ly; Reg[j+1].y2:=y2 end; end;
```

Table 5 presents the results of several steps of the above creation, generating coordinates of the first 12 sectors. Here, sectors 1, 4, 5, 6, and 7 shall be divided by vertical cut line, and sectors 1, 2, 3, and 8-12 by horizontal line.

<table>
<thead>
<tr>
<th>№ of sector</th>
<th>Coordinates of sector angles</th>
<th>№ of sector</th>
<th>Coordinates of sector angles</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$x_1$</td>
<td>$y_1$</td>
<td>$x_2$</td>
</tr>
<tr>
<td>1</td>
<td>1000</td>
<td>1000</td>
<td>4200</td>
</tr>
<tr>
<td>3</td>
<td>2600</td>
<td>1000</td>
<td>4200</td>
</tr>
<tr>
<td>5</td>
<td>1000</td>
<td>2200</td>
<td>2600</td>
</tr>
<tr>
<td>7</td>
<td>2600</td>
<td>2200</td>
<td>4200</td>
</tr>
<tr>
<td>9</td>
<td>1800</td>
<td>1000</td>
<td>2600</td>
</tr>
<tr>
<td>11</td>
<td>1800</td>
<td>2200</td>
<td>2600</td>
</tr>
</tbody>
</table>

Practice of employing the algorithm for compact placement shows that an exchange of fragments with equal attractions in neighboring positions, when they are assigned to the next group, is the reason for possible worsening of the degree of connectivity of fragments inside the formed sectors. It requires introduction of certain changes into the strategy of dichotomous division, with the resultant more differentiated estimation of attraction criterion for the stated fragments. To this end, 4 fragments placed in the nodes of minimum quadrate of lattice shall be defined in some configuration. They shall be called a group. Each group shall be either intrinsic, or outer. Each intrinsic group may have 2-4 respective outer groups, as it is shown in Fig. 11.

![Fig.11. Placement of elements-poles](image-url)
With respect to fragments of intrinsic group, fragments of the outer group are actually the temporarily fixed fragments (reference ones) at the subsequent step of dichotomous division, prior to which the resultant length of connections shall be recomputed in each alternate permutation within an intrinsic cycle of placement. The total number of assignment of elements in the positions of intrinsic group will \(4! = 24\) alternatives here, and enumeration of these alternatives shall be made only if, as it is stated above, in computing the resultant criteria of attractions, fragments with equal attractions in the neighboring positions were obtained. In terms of implementing the described procedure using computer, it is wise to assume that the fours of reference fragments are arranged so as their centers are located at a distance of a knight’s move, as it is shown with black quadrates in Fig. 12.

\[\text{Fig. 12. Placement of intrinsic groups and groups-poles}\]

This strategy will exclude potential repeated (and extremely undesirable) movement in the course of dichotomous division of fragments that belong to the used outer groups.

**Computational experiment on the model problem.** Since the number of VLSI fragments may reach hundreds and thousands, to prepare initial data in selecting the model problem, a computer-aided system of designing electronic modules Altium Desiner–2018 was chosen. Up to now, this system, as most of the known CADSs, does not have the appropriate means for automatic placement of elements within a patchboard, although there are quite well-developed tools for interactive arrangement here.

Implemented in the paper package of applied software for placement is designed to work with two types of discrete working area. The first discrete working area is pointwise with \(64 \times 64 = 4096\) discretes. Here, during placement no consideration is taken to dimensions of fragments, i.e. their superposition against each other is allowed. Software for placement enables, hence, to automatically place more than 4000 fragments of VLSI circuits. Dimension of the second discrete working area is \(256 \times 256\), which allows to take into account the size of fragments in the course of final placement. Software is made compatible with Altium Designersystem on the basis of the database of the latter, which format in the text form is accessible and uploaded by developers of CADS Altium Designer to Internet.

The model problem as such is a problem of placing connection of two- and three—terminal fragments of VLSI circuits, bound by test electrical diagram, with the number of test fragments of VLSI and circuits connecting them, alternating from one experiment to the other.

**Results**

A scheme for the model problem is chosen in such a way that test tightly coupled groups of fragments (5 in a group) follow one by one within it. It has been made deliberately to demonstrate effectiveness of software operation, which finds these groups and uniformly distributes them across a crystal. It is clearly demonstrated in Figure 13 further obtained after placement of model fragments in discrete working area \(16 \times 16\). Figures 14 and 15 show the structure of VLSI fragments’ connections in the amount of 1250 and 2500 units, respectively, placed within \(64 \times 64\) area.

Analyzing the structure of connections shown in these figures, clearly demonstrates the fact of integrating the fragments into tightly coupled groups, since most of these connections are short. Besides, the structure of connections itself became simplified: both their length, and the number of intersections decreases.
Fig. 13. Topology obtained after placing fragments within 16 x 16 area

Fig. 14. Structure of placement connections of 1250 fragments within $64 \times 64$ area
Fig. 15. Topology obtained after placing fragments within $64 \times 64$ area

An important feature of the suggested algorithm is its ultra fast-response, since no search procedures are used in it. Table 6 provides data, obtained from the computer-based experiment and presenting dependence of counting time on the number of VLSI fragments that take part in placement. The table shows that if VLSI circuit has approximately one thousand of fragments, the software program will place them within several seconds. However, if the number of fragments is close to 2500, it will take minutes to place them.

The upper curve in the figure demonstrates dependence of time for placement on the number of VLSI fragments, constructed by the results of several computational experiments, in each of which the number of placed fragments was on the increase by 300 pieces.

![Dependence of time for placement on the number of VLSI fragments](image)

Fig.16. Dependence of time for placement on the number of VLSI fragments
The authors were carrying out a supplementary experiment that involved the model problem with 8 fragments, aimed at comparing the algorithm of dichotomous division placement, implemented in the paper, with the exhaustive algorithm [10]. The results have come as a surprise: placement obtained using dichotomy, has shown a value of the total length of connections, greater by 5% on the average, however, the number of really separated connections turned out to be greater than that for placement obtained by the full enumeration method.

Discussion of results

Table 7 presents a general pattern of growth of the time for placement, as the number of placed VLSI fragments increases. It should be noted, however, that the form of the curve on the graph, presented in Fig.16 and constructed using data of Table 6, is non-linear.

This dependence appears to be linear and the curve should pass in much the same manner, the lower line in figure 16 passes. Indeed, when the number of VLSI fragments that take part in placement, is \( n = 2 \), the number of dichotomous division steps ( \( N \) )is 1, when the number of fragments \( n = 4 \), this number will be 3. Then, the number of division steps will increase the way that it is shown in Table 7, i.e. in a linear manner (it is easy to verify that the line constructed using the data from Table 7 will pass at an angle 45°).

\[
\begin{array}{|c|c|c|c|c|c|c|c|c|c|}
\hline
N & 1 & 3 & 7 & 15 & 31 & 63 & 127 & 255 & 511 & 1023 & 2045 \\
\hline
\end{array}
\]

Table 7

To explain the created paradox, tables 6 and 7 and figure 8 shall be used, where the results are presented of calculating criteria of attracting fragments when dividing sectors by cut lines \( S_1, S_2, \ldots S_7 \). The point is that division of fragments is not finished immediately after defining by the algorithm a dichotomous division of sectors, formed by the subsequent cut line. A mechanism of the algorithm work is such as if a random sector to be divided has \( m \) fragments, its division into 2 sectors shall be made not in one step, but in \( \frac{m}{2} \) steps. At each of these steps not all \( \frac{m}{2} \) fragments are finally distributed, but only two of them, which have the extremum values of attraction criteria to the left and right sectors. Furthermore, as it is shown in Table 2, immediately after determining this “extremum” pair of VLSI fragments, each of them acquires a status of the corresponding pole. The said shall be clarified by the example of dividing the central sector of vertical (figure 5) cut line. This division shall be made in two steps. Column \( M_1 \) of table 2 contains signs of poles prior to starting division of the central sector, and column \( M_2 \) contains signs of poles prior to the second step. Since in the first step, an “extremum” pair (fragments 9 and 10) has its positions in sectors, in the second step they obtained a status of poles, as is presented in column \( M_2 \) of Table 2.

The described procedure for \( \frac{m}{2} \)-step-wise separation of fragments at the subsequent stage of passing a cut line, dividing the next sector of \( m \) fragments by halves, is the reason for increasing the time of division and deviation of the graph from linear dependence. And the more the total number of placed fragments, the more this deviation is. This procedure is implemented in block 6 of the above-described general algorithm of dividing sectors into parts. Quadratic nature of dependence shown in Fig.16 and constructed using data from Table 6, is proved by the following contemplations. 2 new sectors with \( \frac{n}{2} \) fragments in each of them are formed in the first step, hence, the number of place-
moment iterations is $\frac{n}{2}$. 4 new sectors with $\frac{n}{4}$ fragments in each of them are formed in the second step, hence, the number of placement iterations in each of them is $\frac{n}{4}$ and, since 2 sectors shall be divided, then the number of iterations in the second step will again amount to $\frac{n}{2}$. 8 new sectors with $\frac{n}{8}$ fragments in each of them are formed in the third step, hence, the number of placement iterations in each of them is $\frac{n}{8}$ and, since 4 sectors shall be divided, then the number of iterations here is $\frac{n}{2}$. Using the same line of reasoning, the final formula shall be obtained, which specifies complexity (S) of computational procedure for algorithm of dichotomous division placement, implemented in the paper:

$$S = \frac{n}{2} + 2 \times \frac{n}{4} + 4 \times \frac{n}{8} + \ldots + n \times \frac{n}{2n} = \sum_{i=1}^{n} \frac{n}{2} = \frac{n}{2} \cdot (n-1) \cdot \frac{n}{2}$$

As is seen, the obtained dependence is quadratic that is proved by a graph in Fig.16. The obtained and described previously in section “Results” effect of increasing the number of really separated conductors for placement using dichotomous method is likely to be explained by the fact that the suggested algorithm at each step of division actually optimizes the structure of connections, and the effect of decreasing the total length of connections is secondary. Indeed, when describing the principle of its work and examples, this criterion was not mentioned anywhere.

**Conclusion**

An algorithm is suggested of placing fragments of very large scale integration circuits by the method of dichotomous division, and its step-wise description is given. The work of algorithm is demonstrated on several numerical examples. The results are presented of comparing the suggested algorithm with hybrid algorithm of fractal aggregation that works jointly with genetic algorithm. A comparison, which were performed using one and the same model example, has shown that the proposed algorithm allows to obtain placement of better quality in terms of decreasing total length of connections. Computational complexity of the suggested algorithm was theoretically determined that is in compliance with the data of computational experiment. A graph of dependence of counting duration on the number of the placed VLSI fragments. Computational experiment that involved a model problem has shown that the total time for placement of the two and a half thousands of VLSI fragments using algorithm of dichotomous division amounts to 600 sec. The results of computational experiment were demonstrated on the model problem. They illustrated that the structure of connections between fragments is drastically simplified.

**References**


8. Ginzburg B.D. Algorithm of on-board placingmodules// Obmen opyтом в радиопромышленности/Sharing ex-
perience in radio industry. – 1972. - Issue 4. 31-33. 87 / Гинзбург Б.Д. Алгоритм размещения модулей на плате // Обмен опытом в радиопромышленности. - 1972. - Вып.4. 31-33. 87.


