

Hybrid evolutionary algorithm of planning VLSI

V.B. Lebedev*, A.N. Samoylov**

Nekrasovsky lane, GSP-17A, Taganrog, 347928, Russia

*(+7) 8634 37-17-87, *lebvlad@rambler.ru, **samoylov@tsure.ru*

Abstract

Hybrid evolutionary algorithm and technique of planning VLSI are described. Process of slicing floorplan synthesis is broken into two connected stages- synthesis of the slicing tree structure and floorplan formation by aggregation method. During genetic search the decisions specifying structure of slicing tree are exposed to evolutionary changes. On each step of genetic search the plan is under construction by application to this decision of aggregation procedure constructed on the basis of mechanisms of collective alternative adaptation. Two approaches to reduction of total area of plan are used in work. At the first approach the sizes of modules are fixed. It is possible to change modules orientation. At the second approach the sizes of modules can change according to restrictions. The linear estimation of time complexity of decoding procedure, and also structure, at which chromosomes and the genes located in the same loci, are homologous, simplifies genetic operators performance procedures and the genetic search organization. Experimental research has shown that algorithms at teamwork within the framework of the common approach give high quality.

Keywords: Planning, VLSI, genetic algorithm, collective adaptation.

INTRODUCTION

Planning VLSI consists in accommodation on a chip of the modules which are having the set area and not having the fixed sizes [1]. Modules and a chip have the form of rectangular. These modules are to be placed in rectangular area in non-lapping manner. As a result of planning are solved at once two problem: the relative positioning of modules from each other is defined, i.e. their accommodation, the sizes of each module also are fixed. As a result of planning the plan of a chip representing covering rectangular, divided by horizontal and vertical segments on non-lapping rectangular blocks in which it is necessary to place corresponding modules is under construction. The basic purpose of optimization is minimization of a plan total area.

The problem of planning is classified as NP problem. For the last few years various approaches to the decision of planning problem have been offered. These approaches can be classified as follows [1,2,4,5,6,7,8]: linear and square-law programming; simulated annealing; based on restrictions; force directed approach; based on geometrical dualization lists of connections; hierarchical methods from top to down and from below - upwards; a clustering method; numerical optimization techniques genetic algorithms (GA) [9,10,11,12]; on the basis of search adaptation [12,13], etc. The analysis of existing approaches to the decision of this problem has

shown, that the approaches based on methods of evolutionary modeling [14, 15] are popular in determining the optimal decisions.

These methods differ from each other, of both statements and models. Insufficiently full account of specificity of a considered problem and absence of adequate ways of representation make a problem of the perfect evolutionary models construction make actual.

The planning problem is formulated as follows [1]. There is a set of modules $M = \{m_i \mid i=1,2,\dots,n\}$. A module can be a fixed type or a flexible type. A fixed module has fixed height and width. A fixed module may be rotated by $+90^\circ$. Each flexible module is characterized by the three numbers $\langle S_i, l_i, t_i \rangle$ where S_i – is the area of the module m_i , and parameters l_i and t_i set the bottom and top border of aspect ratio h_i / w_i , i.e.

$$l_i \leq h_i / w_i \leq t_i, \quad (1)$$

where h_i – is a module height, w_i – is a module width.

The plan for set of modules of M represents rectangular R cut by vertical and horizontal lines on set of areas u_i , in each of which the module m_i is located accordingly. Otten first proposed a binary-tree representation for slicing floorplan design [16]. In our work we use a slicing floorplan with a slicing structure.

Slicing floorplans are those that can be recursively bisected by horizontal and vertical cut-lines down to single blocks. A slicing structure can be represented by a binary tree whose leaves denote modules, and internal nodes specify horizontal or vertical cut lines. On figure 1 the slicing floorplan and the corresponding slicing tree is shown.

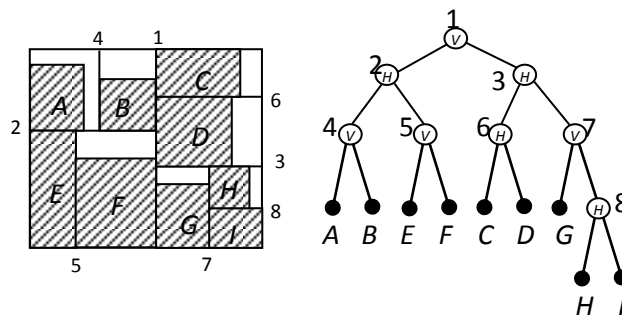


Figure 1. A slicing floor plan and the corresponding slicing tree.

On a slicing tree (figure 1) the nodes corresponding to cuts are marked by figures, and V is a vertical cut, and H is a horizontal cut. Letters mark the leaves corresponding to areas. Each area u_i , intended for accommodation of the module m_i , has the sizes x_i and y_i . It is obvious, that the area sizes should correspond to the following restrictions.

$$S_i = w_i \cdot h_i, S_i \leq x_i \cdot y_i, h_i \leq y_i, w_i \leq x_i. \quad (2)$$

at simultaneous observance of restrictions (1).

The purpose of optimization is minimization of the total area of plan R, at observance of restrictions (1), (2).

Previous applications of GA [17,18,19] to the floorplanning involve floorplan with fixed module, use complexity crossover and mutation mechanisms for generating a valid offspring and are therefore limited in application. We develop a GA for floorplan area optimization by applying new principles and ways of coding and decoding of chromosomes for representation of the trees. A module can be a fixed type or a flexible type. The chromosomes are homologous that excluding incorrect decisions and simplifies performance genetic operators and the genetic search organization. In Hybrid evolutionary algorithm of planning VLSI the process of tree structure synthesis and floorplan formation by aggregation method are divided. During genetic search the decisions specifying structure of slicing tree are exposed to evolutionary changes. The structure of slicing tree is given by three chromosomes $H = \{H1, H2, H3\}$. On each step of genetic search the plan is under construction by application to this decision of aggregation procedure constructed on the basis of mechanisms of collective alternative adaptation.

Formation of the floor plan by blocks aggregation method from given slicing tree

In common case in genetic algorithm the decision is encoded by set H consisting of four chromosomes $H = \{H1, H2, H3, H4\}$. Chromosome H1 contains the information on a labels of slicing tree leaves which correspond to the modules. Chromosome H2 contains the information on slicing tree structure. Chromosome H3 contains the information on types of cuts (H or V). Chromosome H4 contains the information on value of aspect ratio h_i/w_i . At the fixed sizes of modules chromosome H4 contains the information on orientation of modules.

On the basis of this information the slicing floorplan design is carried out by consecutive binary aggregation of areas on a slicing tree, starting from leaves of a tree. To each internal node of a slicing tree there will correspond the area received as a result of aggregation of a subtree, having this internal node as a root. Let us consider, that a cut with number i, cuts the node d_i (area u_i). In the beginning of aggregation each node d_i , being the leaf of a slicing tree, will correspond to the area u_i with the sizes $x_i = h_i$, $y_i = w_i$ equal to the sizes of the module m_i . Each internal node d_i of a slicing tree will correspond to the area u_i which is formed by slicing subtree aggregation, having node d_i as a root.

Let nodes d_i and d_j are children nodes of node d_k and let for areas u_i and u_j corresponding d_i and d_j the bottom borders of their sizes (x_i, y_i) , (x_j, y_j) are determined.

Process of binary aggregation represents merge of areas u_i and u_j , formation of area u_k , definition of the sizes for u_k and the new sizes for u_i and u_j . We shall enter two infix operators V and H. Record $u_k = u_i H u_j$ means, that areas u_i and u_j merge on a horizontal in one area u_k . If $u_k = u_i V u_j$, areas u_i and u_j merge on a vertical.

In process of binary aggregation of areas on a slicing tree will be received the expressions determining the sizes of a rectangular R.

Let us designate through $\max(x_1, x_2)$ the maximal value from x_1 and x_2 . At merge on a horizontal $x_k = \max(x_i, x_j)$; $y_k = y_i + y_j$; x_1 and x_2 will have the size equal $\max(y_i, y_j)$. At merge on a vertical $x_k = x_i + x_j$; $y_k = \max(y_i, y_j)$; y_1 and y_2 will have the size equal $\max(y_i, y_j)$.

Let for an example (fig. 1) at the fixed sizes of modules (A, B, C, D, E, F, G, H, I) after metrization the plan with the modules placed in areas looks like it is submitted on figure 1. We shall distinguish the sizes (h_i, w_i) of the module m_i from the sizes (x_i, y_i) of area u_i in which the module (block) m_i is placed. The sizes of areas, according to consecutive aggregation, are defined as follows:

1. $u_8 = m_H \text{ H } m_I$. $x_8 = \max(w_H, w_I) = w_I$; $y_8 = h_H + h_I$.
2. $u_7 = m_G \text{ V } u_8$. $x_7 = w_G + x_8 = w_G + w_I$;
 $y_7 = \max(h_G, y_8) = y_8 = h_H + h_I$.
3. $u_6 = m_C \text{ H } m_D$. $x_6 = \max(w_C, w_D) = w_C$; $y_6 = h_C + h_D$.
4. $u_5 = m_E \text{ V } m_F$. $x_5 = w_E + w_F$; $y_5 = \max(h_E, h_F) = h_E$.
5. $u_4 = m_A \text{ V } m_8$. $x_4 = w_A + w_B$; $y_4 = \max(h_A, h_B) = h_A$.
6. $u_3 = u_6 \text{ H } u_7$. $x_3 = \max(x_6, x_7) = x_7 = w_G + w_I$; $y_3 = h_C + h_D + h_H + h_I$.
7. $u_2 = u_4 \text{ H } u_5$
 $x_2 = \max(x_4, x_5) = x_5 = w_E + w_F$; $y_2 = y_4 + y_5 = h_A + h_E$.
8. $u_1 = u_2 \text{ V } u_3$. $x_1 = x_2 + x_3 = w_E + w_F + w_G + w_I$.
 $y_1 = \max(y_2, y_3) = y_3 = h_C + h_D + h_H + h_I$.

Let us name the expressions determining the sizes of a rectangular received as a result of consecutive aggregation - calculated. In our case calculated expressions look like:

$$x_1 = w_E + w_F + w_G + w_I; \quad y_1 = h_C + h_D + h_H + h_I. \quad (3)$$

At final definition of the sizes (h_0, w_0) describing rectangular R three situations are possible.

If restriction is $-l_0 \leq y_1/x_1 \leq t_0$ then $h_0 = y_1$, and $w_0 = x_1$, i.e. the sizes of rectangular R coincide with the sizes u_1 .

If $l_0 > y_1/x_1$, then $w_0 = x_1$, and $h_0 = l_0 \cdot x_1$.

If $y_1/x_1 > t_0$, then $h_0 = y_1$, and $w_0 = y_1/t_0$.

The total area SR of plan R is $h_0 \cdot w_0$.

In the received expressions it is possible to reduce the total sizes (h_0 , w_0), having reduced first of all the sizes of the modules being components of expressions, determining sizes h_0 , and w_0 .

With a view of labour input reduction of in work the process of the plan synthesis is broken into two connected stages - synthesis of slicing tree and formation of the plan by blocks aggregation method from given slicing tree [8,13].

During genetic search the decisions set given by three chromosomes $H = \{H_1, H_2, H_3\}$ are exposed to evolutionary changes. The plan is under construction by application of aggregation procedure to this decision. The estimation of the plan constructed with the help of aggregation procedure, is considered as an estimation of the decision.

Two approaches to slicing floorplan design are used in work.

At the first approach the sizes of modules are fixed. It is possible to change modules orientation. For the given slicing tree the modules orientations providing the minimal values of parameter $F = SR$ are found at aggregation.

At the second approach the sizes of modules can change according to restrictions (1), (2). For the set slicing tree the sizes of the modules corresponding to restrictions (1), (2) and providing the minimal value of parameter $F = SR$ are found at aggregation.

Mechanisms of genetic search

As it was already specified above a slicing tree corresponds to the plan of a crystal based on "guillotine cut". The leaves of the slicing tree are the nodes corresponding to blocks. To set a slicing tree, first, to set structure of this tree, i.e. sequence of binary cuts, second, for the internal nodes of a slicing tree corresponding to cuts to specify type of cut H or V. Thirdly, to mark trailing nodes of a tree (leaves) with numbers of blocks. We shall designate as $E = \{e_i \mid i = 1, 2, \dots, n\}$ the set of trailing tops of a slicing tree which are situated down in a horizontal line from left to right. It is possible to set a marking of blocks with the help of vector $Q = \langle q_i \mid i = 1, 2, \dots, n \rangle$, where q_i is the block number. For example: for a tree on fig. 1 vector Q looks like: $Q = \langle A, B, E, F, C, D, E, H, I \rangle$. With the help of rearrangement of elements of vector Q it is possible to set any marking of nodes of set E.

The base of genetic algorithm is principles of the coding and decoding chromosomes, genetic operators and genetic search structure based on natural selections ideas form.

Principles of coding and decoding of chromosomes. In work the decision is coded by set H consisting of three chromosomes $H = \{H1, H2, H3\}$ [21]. Chromosome $H1$ contains the information on a marking of set of nodes E . Chromosome $H2$ contains the information on a slicing tree structure. Chromosome $H3$ contains the information on types of cuts (H or V).

Let n – be a number of areas of the plan (number of nodes of set E). Chromosome $H1$ specifying a marking of set of nodes E , looks like $H1 = \{g1i \mid i = 1, 2, \dots, n-1\}$. Each gene $g1i$ can accept any value in an interval from 1 up to $(n+1-i)$.

For example: for $n = 8$; $1 \leq g11 \leq 8$; $1 \leq g12 \leq 7$; $1 \leq g13 \leq 6$; ...; $1 \leq g17 \leq 2$.

Decoding of chromosome $H1$ is made with use of basic vector $B1 = \langle b1i \mid i = 1, 2, \dots, n \rangle$ which number of elements equally n , and their values lay in an interval from 1 up to n . Let for $n = 8$ there is chromosome $H1 = \langle 3, 5, 3, 4, 4, 2, 2 \rangle$, and let there is basic vector $B1 = \langle 1, 2, 3, 4, 5, 6, 7, 8 \rangle$.

We consider under the order genes of a chromosome and according to their values we choose elements in a basic vector and they are written down in the course of selection in vector Q .

Value $g11 = 3$. We choose in $B1$ an element $b1j$ ($j = g11 = 3$, $b13 = 3$) and write down it on the first place of formed vector Q , i.e. $q1 = b13 = 3$.

We delete an element $b13$ from $B1$ and we receive vector $B2 = \langle 1, 2, 4, 5, 6, 7, 8 \rangle$ containing 7 elements. The following $g12$, $g12 = 5$ is selected. We find an element $b25$ of vector $B2$, $b25 = 6$. Hence $q2 = 6$. We delete an element $b25$ from $B2$, we receive vector $B3 = \langle 1, 2, 4, 5, 7, 8 \rangle$. Further:

$13=3$	$33=4$	$3=4$	$B4=$ $\langle 1, 2, 5 \rangle$
$14=4$	$44=7$	$4=7$	$B5=$ $\langle 1, 2, 5 \rangle$
$15=4$	$54=8$	$5=8$	$B6=$ $\langle 1, 2, 5 \rangle$
$16=2$	$62=2$	$6=2$	$B7=$ $\langle 1, 5 \rangle$
$17=5$	$72=5$	$7=5$	$B8$ $= \langle 1 \rangle$

$q8 = b81 = 1$.

In a result we receive vector $Q = \langle 3, 6, 4, 7, 8, 2, 5, 1 \rangle$, specifying a marking of set of nodes E .

Let us consider structure of chromosome $H2$. Wong and Liu in [20] presented a normalized Polish expression to represent a slicing floorplan. We shall enter alphabet $A = \{X, \bullet\}$. Structure of a slicing tree can be set with use of polish

expression for binary tree based an alphabet A, where sign X corresponds to leaves of a slicing tree and sign • - corresponds to internal nodes of a tree (cuts). Polish expression for a tree submitted on fig. 1 looks like: X X • X X • • X X • X X X • • • • .

Process of restoration of a tree according to the polish expression is simple enough. Consistently from left to right the Polish expression is looked through, and letters such as •, corresponding to cuts are found. Each such cut unites two nearest subgraphs formed on the previous steps, located in the polish record to the left of sign •. Let us illustrate convolution process with the help of brackets: ((X X •)(X X •) •)((X X •) (X (X X •) •) •) • .

Let us note the basic properties of the polish expression for the performance of which it is necessary that a slicing tree corresponds to a record.

Let us designate as n_x - number of elements of the polish expression such as X, and as n_{\bullet} - number of elements such as •. For a slicing tree equality $n_x = n_{\bullet} + 1$ is always carried out.

If in the polish expression we make a section to the right of sign • to the left of section the number of signs X is more than number of signs • at least on unit. First sign • in the polish expression (at viewing from left to right) can appear only after two signs X. We shall number positions between signs X, as shown below

XX 1 X 2 X 3 X 4 ... X $N_x - 1$.

The maximal number of signs • which can appear in a position, is equal to a position number. Let us remind, that the common number $n_{\bullet} = n_x - 1$. If the polish expression corresponds to the properties listed above a slicing tree corresponds to it.

Chromosome H2 looks like, $H_2 = \{g_{2i} \mid i = 1, 2, \dots, n_{\bullet}\}$. As a result of a chromosome decoding the polish expression is under construction. Value of a gene g_{2i} changes in limits from i up to n_{\bullet} i.e. $i \leq g_{2i} \leq n_{\bullet}$. Value of a gene specifies number of a position in the Polish expression in which it is necessary to place sign •.

With the help of chromosome $H_3 = \{g_{3i} \mid i = 1, 2, \dots, n_{\bullet}\}$ types of cuts (H or V) are set. Value of a gene is 0 or 1, thus 0 – corresponds to H, and 1 – corresponds to V. The marking of internal node s (cuts type definition) is carried out consistently by way of signs • arrangement in the polish expression.

For example: Let $H_1 = \langle 2, 1, 3, 1 \rangle$, $B_1 = \langle 2, 1, 4, 5, 3 \rangle$, $H_2 = \langle 4, 2, 2, 4 \rangle$ and $H_3 = \langle 1, 0, 0, 1 \rangle$ vector Q corresponding to chromosome H1 looks like: $Q = \langle 1, 2, 3, 4, 5 \rangle$. The Polish expression corresponding H2, looks like X X X • • X X • •. The polish expression modified with account H3 will become: X X X H V X X V H.

Process of decoding (construction of a phenotype) on chromosomes H1, H2, H3 consists in construction of vector Q on chromosome H1 and the modified polish expression on H2, H3. Each of the listed procedures has an estimation of labour input $O(n)$, where n is the number of modules. Thus the general estimation of labour input

is equal $O(n)$. Spatial complexity for one decision also has an estimation $O(n)$. The linear estimation of time complexity of decoding procedure, and also structure, at which chromosomes and the genes located in the same loci, are homologous, simplifies genetic operators performance procedures and the genetic search organization.

Genetic operators. In this paper two types of crossover operators K1 and K2 are used. With help of K1 new decisions are made by a combination of structures of parental pair chromosomes $R1=\{H11,H21,H31\}$ and $R2=\{H12,H22,H32\}$. Parameter Pk1- crossover probability is set. Pairs homologous chromosomes $\{H11,H12\}$, $\{H21,H22\}$ and $\{H31,H32\}$ lists R1 and R2 and with probability Pk1 an exchange of chromosomes of parental pair R1 and R2, is carried out.

With help of K2 new decisions are formed by means of an exchange of genes at homologous chromosomes of parental pair. Parameter Pk1 - crossover probability is set. Pairs of homologous chromosomes $\{H11,H12\}$, $\{H21,H22\}$ and $\{H31,H32\}$ are selected consistently. Consistently, starting with the first, we look through loci of the chosen pair homologous chromosomes. With probability Pk2 the exchange of homologous genes is carried out.

Combined crossover K3 is realized as follows. First on the chosen parental pair R1 and R2 crossover K2 is realized, i.e. the genes exchange is carried out. Affiliated pair R1' and R2' is formed. Further this pair is considered as parental and crossover K1 is applied to it, i.e. the chromosomes exchange is carried out. As a result of consecutive application of crossover K2 and K1 to parental pair R1 and R2 affiliated pair R1'' and R2'' is formed.

Updating K3 consists that before each application of K2 value PK2 ($0 \leq PK2 \leq 1$) is selected casually.

The operator of a mutation is carried out as follows. Parameter Pm- probability of a mutation is set. Loci of chromosomes of the decision are looked through consistently. With probability Pm the mutation of a gene in the current locus is carried out. The mutation consists in acceptance by a gene of casual value from the set range of values for a gene in the given locus.

In a basis of selection as for a choice of chromosomes pair at crossover, and at selection of a population the principle of roulette wheel selection is put. In this scheme, the probability of selection of a member in a population is proportional to its fitness value.

Apparently from the algorithms, realizing crossover and mutations operators, the estimation of their running time looks like $O(n)$. The organization of genetic search procedure at planning. The decision representation in the form of 3 chromosomes gives possibility to use the operator of chromosomes set combining in one decision that means the natural evolution approach. On the other hand, representation of the decision by a set from 3 chromosomes enables the organization

of decisions search in various statements, leaving separate kinds of chromosomes constant during genetic search.

For example: at fixed $H1$, $H2$ to search for the optimum decision only due to change $H2$, i.e. types of cuts (H or V). It is obvious, that fixing of separate chromosomes in some statement results in narrowing of search space, but thus loss of optimum decisions is possible. In this connection the combination is represented by separate statements expedient by search of the optimum decision.

Aggregation on the basis of collective adaptation at planning VLSI with varying orientation of modules

Let us present the initial formulation of a problem as the adaptive system working on the basis of modeling of collective behaviour of automatic devices of adaptation [13].

Each module_ is free to rotate. Orientation of the module m_i at accommodation in area we shall set in parameter o_i . $o_i \in \{1, 2\}$, i.e. for m_i two ways (two orientations) accommodations in area u_i are possible (figure 2). Let us designate as $(h1_i, w1_i)$ the sizes m_i at $o_i = 1$, and through $(h2_i, w2_i)$ the sizes m_i at $o_i = 2$. Let the first orientation there corresponds such arrangement of modules m_i , that $h1_i / w1_i \leq 1$, and at the second orientation $h2_i / w2_i > 1$. There are: $h1_i = w2_i$; $h2_i = w1_i$; $h1_i \leq h2_i$; $w2_i \leq w1_i$.

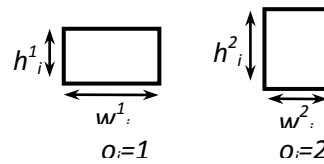


Figure 2. Orientation of the module m_i

Set of modules $M = \{m_i | i=1, 2, \dots, n\}$ with the fixed sizes, a slicing tree $D = \{d_j | j=1, 2, \dots, 2n-1\}$ and orientations of modules $O = \{o_i | i=1, 2, \dots, n\}$ define unequivocally the plan which is under construction with the help of aggregation procedure.

Space of decisions is represented by the decisions distinguished from each other by values of elements of set O , making modules orientation. We shall present search process in decisions space as the adaptive system working in conditions of uncertainty.

On each step of adaptive system work under action of adaptive influences the transition from one node of decisions space to another is carried out, i.e. a choice of alternative values of elements of set O . As objects of adaptation we shall consider modules m_i . Each object can be only in one of two alternative conditions $A1_i$ or $A2_i$. The condition of adaptation object corresponds to the chosen alternative (orientation): $A1_i$ corresponds to the first orientation; $A2_i$ – to the second.

Let for set of modules M with the fixed sizes some initial plan is set, i.e. the slicing tree, orientation of modules is set, according to the chosen alternatives,

aggregation of areas is made and expressions for definition of values of parameters h_0 and w_0 rectangular R are generated. For each m_i the environment will be a set of cooperating with each other and with m_i modules $M_i = M \setminus m_i$. The condition of environment is defined by the orientations chosen for all modules. The estimation of conditions of adaptation object depends both on a condition of environment, and on a condition of adaptation object in the environment.

Let s_1 and s_2 are the sizes of the module m_i and $s_1 > s_2$. If any size of the module m_i is a part of expression for w_0 (or h_0) then s_2 will be more preferable

The local purpose of adaptation object m_i - is to accept such orientation at which the structure of calculated expressions for h_0 or w_0 includes the size of the smaller side of the module m_i .

The global purpose of collective of adaptation objects, i.e. set of modules M , - is to achieve such condition at which value of criterion F has the minimal value.

Let us enter two parameters $\langle Pix, Piy \rangle$ for an estimation of a condition of object of adaptation (i.e. the module m_i) in the environment. We shall define possible values of these parameters.

$Pix = 1$, if the expression for w_0 include the size w_i of the module m_i ;

$Pix = 0$ otherwise.

$Piy = 1$, if the expression for h_0 include the size h_i of the module m_i ;

$Piy = 0$ otherwise.

Four combinations of values of parameters $\langle Pix, Piy \rangle$ are possible: $\langle 0, 0 \rangle$, $\langle 0, 1 \rangle$, $\langle 1, 0 \rangle$, $\langle 1, 1 \rangle$.

If $Pix = 1$ alternative A_{2i} will be more preferable since $w_{2i} < w_{1i}$, and, hence the smaller value will enter, in expression for w_0 .

If $Piy = 1$ alternative A_{1i} will be more preferable since $h_{1i} < h_{2i}$, and, hence the smaller value will enter, in expression for h_0 .

The second way of definition of values $\langle Pix, Piy \rangle$ is possible.

$Pix = 1$ if $w_i = x_i$, i.e. the width w_i of the module m_i is equal to width x_i of area u_i ;

$Pix = 0$ otherwise.

$Piy = 1$ if $h_i = y_i$, i.e. the height h_i of the module m_i is equal to height of area u_i ;

$Piy = 0$ otherwise.

At values $Pix = 1$ or $Piy = 1$ the same alternatives will be preferable, as at the first way as their realization results or can lead to minimization of the area sizes, and also the plan as a whole consequently.

For realization of the adaptation mechanism of to each module m_i the automaton of adaptation AA_i with two groups of conditions $\{C1_i, C2_i\}$, corresponding to two alternatives $A1_i$ and $A2_i$ is put in conformity. The number of conditions in group is set by parameter Q_i named depth of memory. On an input of the automaton of adaptation AA_i the signal "encouragement" or "punishment" moves depending on a condition of adaptation object (the module m_i) in the environment. On figure 3 the graph - circuit of transitions of the automaton of adaptation is shown. Transitions under action of a signal "encouragement" are marked (+), transitions under action of a signal "punishment" are marked(-).

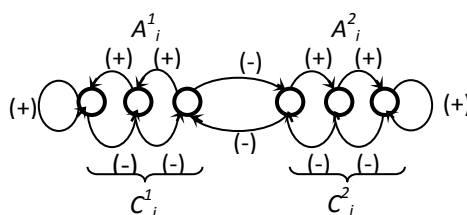


Figure 3. The graph - circuit of transitions of the automaton of adaptation

The technique of managing signals development consists in the following.

If the preferable alternative, according to values of pair parameters $\langle Pix, Piy \rangle$, coincides with the alternative realized at present the signal of encouragement is developed, otherwise - a signal of punishment. Let us note, that in a case when the pair parameters $\langle Pix, Piy \rangle$ have values $\langle 0,0 \rangle$ or $\langle 1,1 \rangle$ preferable alternative does not exist. Therefore in a case $\langle 0,0 \rangle$ the managing signal is not developed at all. And in a case $\langle 1,1 \rangle$ with probability P the signal of punishment is developed. Development of a signal of punishment in such situation promotes an output from a local optimum. Concrete value P can be selected as a result of experimental research.

Rules of development of managing signals are resulted in table 1. Here EA – is a preferable alternative, RA – is the realized alternative, U – is a managing signal.

The local purpose of each adaptation object (the module m_i) – is to achieve such condition at which the preferable alternative coincides with realized one.

Work of adaptive system on each iteration is carried out for four steps. On the first step, according to calculated expressions for definition of the sizes w_0 and h_0 of rectangular R , for each module m_i values of pair parameters $\langle Pix, Piy \rangle$ are defined. On the second step for each adaptation automaton, according to values of pair parameters $\langle Pix, Piy \rangle$ and a condition of the automatic device of adaptation ($C2_i$ or $C1_i$), managing signals ("encouragement" or "punishment") are developed. On the third step under action of a managing signal transition in adaptation automaton is carried out. On the fourth step, according to conditions of adaptation automaton, alternatives of modules m_i , are realized i.e. their corresponding orientations are selected. Then with the help of aggregation procedure the plan is under constructed,

expressions for definition of the sizes of parameters w_0 and h_0 , and the sizes of all areas are defined.

Table 1.

ix	iy	A	RA	U
		1i	A1i	(+)
		1i	A2i	(-)
		2i	A1i	(-)
		2i	A2i	(+)
		o	A1i or A2i	No
		o	A1i or A2i	(-) with P

Algorithm running time on one iteration has an estimation $O(n)$, where n – is the number of blocks.

5. Aggregation on the basis of collective adaptation at planning VLSI with the flexible sizes of modules

Let us consider the second approach to construction of the plan, connected with an opportunity of change of the modules sizes according to restrictions (1), (2).

On each iteration of work of adaptive system under action of adaptive influences there is a change of the modules sizes.

Object of adaptation and environment are the same, as at the first approach. Parameters and a technique of an estimation of a condition of adaptation object are similar the ones considered above. The condition of adaptation object in the environment is defined by pair $\langle Pix, Piy \rangle$. However the alternatives determining adaptation object behaviour, structure of the adaptation automaton and mechanisms of transitions in it, have their own specificity.

For realization of the adaptation mechanism to each module m_i the adaptation automaton AA_i with three groups of conditions $\{C1i, C2i, C3i\}$, corresponding to three alternatives $A1i, A2i, A3i$ (figure 4) is put in conformity

Adaptation object behaviour in the environment is regulated by three alternatives $A1i, A2i, A3i$.

Alternative A1i correspond decreases of w_i and increases of h_i accordingly at the module m_i . The restrictions $l_i \leq h_i / w_i \leq t_i$, and $h_i \cdot w_i = S_i = \text{const}$ should be observed. Parameter δ , of changes of the module size is set. New values h^*_i and w^*_i , at realization of alternative A1i, are defined as follows:

If $h_i > w_i$, then $h^*_i = h_i - \delta$, and $w^*_i = S_i / (w_i - \delta)$.

According to the set restrictions it is necessary, that $l_i \leq h^*_i / w^*_i \leq t_i$.

After substitution h^*_i and w^*_i we shall receive $l_i \leq (h_i - \delta)2/S_i \leq t_i$. From here parameter δ should satisfy to an inequality:

$$h_i - \sqrt{l_i S_i} \leq \delta \leq h_i - \sqrt{t_i S_i}.$$

If $h_i < w_i$, $w^*_i = w_i + \delta$, and $h^*_i = S_i / (w_i + \delta)$. In this case parameter δ should satisfy to an inequality

$$l_i \leq S_i / (w_i + \delta)2 \leq t_i.$$

It is made for changes of overall dimensions of the module (increase or reduction) to be not more, than on size δ .

Alternative A3i correspond decreases of h_i and increases of w_i accordingly at the module m_i .

If $h_i > w_i$, then $h^*_i = h_i + \delta$, and $w^*_i = S_i / (h_i + \delta)$.

If $h_i < w_i$, then $w^*_i = w_i - \delta$, and $h^*_i = S_i / (w_i - \delta)$.

At the task of alternative A2i the sizes h_i and w_i of module m_i do not changes.

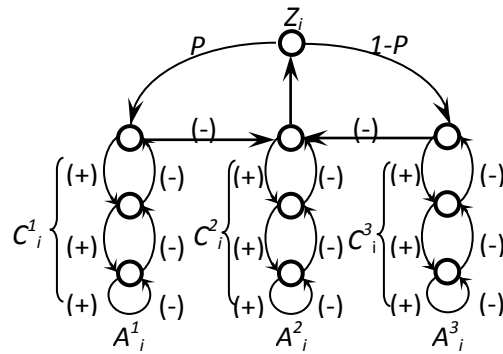


Figure 4. The adaptation automaton with three groups of conditions

The mechanism of development of a managing signal ("encouragement" or "punishment") is based on the following reasons.

If one of parameters of pair $\langle P_{ix}, P_{iy} \rangle$ is equal to unit it is expedient to reduce w_i (for a case $\langle 1, 0 \rangle$), or h_i (for a case $\langle 0, 1 \rangle$). If AA_i are in that group of conditions to which there corresponds the alternative providing similar reduction the signal "encouragement" is developed, otherwise - a signal "punishment".

If the pair looks like $\langle 1,1 \rangle$ then irrespective of, the fact in what group of conditions is AA_i , the signal "punishment" is developed with probability $P=0.2$. Parameter P is changeable and can be picked up as a result of experimental research. Probability development of a signal "punishment" for a case when the pair looks like $\langle 1,1 \rangle$, is intended for an output from a local optimum.

If the pair looks like $\langle 0,0 \rangle$ and AA_i there is in group $C2_i$ the signal "encouragement" is developed, otherwise a signal "punishment".

Transitions AA_i from group $C2_i$ in group $C1_i$ or $C3_i$ are carried out as follows. First at an output from group of conditions $C2_i$ the adaptation automaton passes in intermediate condition Z_i . During presence AA_i in group $C2_i$ the number of punishments α_i received at $P_{ix}=1$, and also the common number of punishments β_i is counted up. Then after transition from $C1_i$ in Z_i at once with probability $P=\alpha_i/\beta_i$ transition in $C1_i$, and with probability $1-P$ in $C3_i$ is carried out.

In table 2 rules of development of managing signals U are resulted.

Table 2

ix	iy	Condition AA_i	U
		$C3_i$	(+)
		$C1_i$ or $C2_i$	(-)
		$C1_i$	(+)
		$C2_i$ or $C3_i$	(-)
		$C2_i$	(+)
		$C1_i$ or $C3_i$	(-)
		$C1_i, C2_i, C3_i$	(-) with probability P

0	M	Decision quality			Generation number		
		pt	5%	5%	in	ax	verage
	.2	1	9	0	0	76	48

The local purpose of object of adaptation (m_i) - to achieve the module conditions at which the parameters $\langle P_{ix}, P_{iy} \rangle$ looks like $\langle 1,1 \rangle$, i.e. when the sizes

(w_i, h_i) of the module m_i coincide with the sizes (x_i, y_i) of area u_i in which m_i is placed.

The global purpose of collective - to achieve a condition with the minimal value of criterion F . Work of adaptive system on each iteration is carried out by four steps.

On the first step for each object values of the two $\langle P_{ix}, P_{iy} \rangle$ are defined. On the second step managing signals ("encouragement", "punishment") are developed. On the third step transitions in adaptation automaton are carried out under action of managing signals.

On the fourth step the alternatives corresponding to conditions of automatic devices of adaptation are realized. The sizes of separate modules change. Further with the help of convolution the plan is being constructed. Expressions for the sizes w_0 and h_0 and the sizes of all areas are defined. Value F for the new decision is calculated.

Experimental research of the given algorithm has shown, that quality of the decision depends on parameter δ . It is expedient to use variable value $\delta(t)$, decreasing during work of adaptive system.

Experimental research of the developed algorithms

Carrying out of experimental research had two purposes: research of evolutionary adaptive search mechanisms for problems being solved and research of the suggested algorithms efficiency.

Research of genetic search of mechanisms. For carrying out of research test examples on 30, 40, 60, 90, 110 blocks have been synthesized. For a finding of the best combination of such parameters as probabilities of a mutation and crossover. P_M and P_K , the size of a population M and generations number T , and also for a choice of sequence and genetic operators type experimental research was carried out as follows. For each structure parameter P_K was fixed at the beginning and parameters P_M, M changed. Then at fixed values of P_M, P_K changed. For each fixed set of parameters a series of experiments was carried out. Results of experimental researches for fixed value $P_K = 0.4, P_M = 0.2$ are resulted in table 3.

Table 3. $P_K = 0.4$

At carrying out of tests for each experiment number of generation after which improvements of an estimation were not observed was fixed. In each series of tests were defined the minimal, maximal and average value of generation number after which improvements of an estimation were not observed. Results are presented in the table in columns "Min", "Max" and "Average". For each series of tests the best decision was defined. Then the number of tests in series at which the optimum decision (column "Opt") has been received, number of tests at which the decision differed from optimum less, than on 5 % (a column " $< 5\%$ ") and number of tests at which the decision differed from optimum more than on 5 % (a column " $> 5\%$ ") was

fixed. As a result of the analysis of results it has been established, that the best combination is $P_M = 0.2$; $P_k = 0.4$.

On figure 5 for the second example the average dependence of quality of the decision on generations number is shown at values of parameters $P_M = 0.2$, $P_r = 0.4$, $M = 50$.

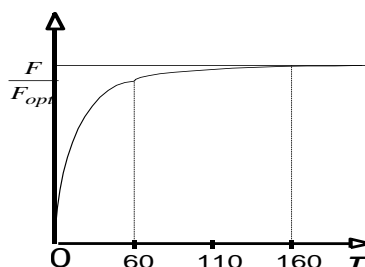


Figure 5. Dependence of quality of the decision on generations number

The diagram shows, that on 150 generations the decision close to optimum is achieved.

Research of collective alternative adaptation mechanisms. At carrying out of experimental researches of adaptive algorithm of planning of crystal VLSI with the floating sizes of modules the following purposes - search of the best combination of values of managing parameters such as q-depth of memory AA, T - number of iterations were pursued.

On the basis of the analysis of researches results it is established, that peak efficiency of adaptive search is provided at values of managing parameters: $q=2$, $T=60$.

Researches have shown, that running time algorithm on one iteration has an estimation $O(n)$, where n – is a number of blocks.

Results of experiments of the developed algorithm were compared on MCNC tests [22] to algorithms [9,10,11,17,19]. In comparison with initial conditions, about 40 % reduction of a microcircuit area are received. The submitted algorithm of aggregation on the basis of collective adaptation in a combination to a genetic synthesis algorithm of a slicing tree finds decisions on the area surpassing results of compared algorithms on the average on 5-10 % besides this it optimizes length of connections. As it is necessary to note, that experimental running time of the submitted algorithm on one iteration at the fixed values of managing parameters - $O(N)$, and running time of existing algorithms makes $O(N^2)$. Experiments have shown, that at the big dimensions time parameters of the developed algorithm surpass parameters of compared algorithms.

CONCLUSION

For the decision of a problem of VLSI planning the approach based on complex use of basic principles of evolutionary and alternative adaptation - self-training and genetic search is used

Two variants of the task of values of the sizes of modules - fixed and flexible are considered, that expands sphere of possible application of the developed algorithms.

In view of specificity of a problem of VLSI planning within the framework of each of considered statements objects of collective adaptation are determined, mechanisms of alternative search adaptation that has allowed for the decision of a planning problem to develop adaptive search procedure are developed and modernized.

New principles and ways of coding and decoding of chromosomes for representation of the trees are developed, the excluding incorrect decisions distinguished by simplicity and linear estimations of time and spatial complexity that simplifies use of genetic operators and allows to use updatings of the genetic operators close to natural.

Multichromosomal representations of decisions have allowed to create hierarchical structures of genetic operators that enables to organize purposeful search.

The submitted algorithm finds decisions on the area surpassing results of existing algorithms, with smaller time expenses for problems of the big dimension besides it optimizes connections length.

REFERENCES

- [1] Naveed Sherwani. Algorithms for VLSI Physical Design Automation. Kluwer academic publishers. Boston /Dordrecht/ London. 1995.
- [2] M. Sarrafzadeh and C. K. Wong. An Introduction to VLSI Physical Design. New York: McGraw Hill. 1996.
- [3] A.B.Kahng "Classical Floorplanning Harmful", ISPD 2000, pp. 207-213.
- [4] J.Cong et al. "Microarchitecture Evolution With Physical Planning", DAC, 2003, pp. 32-35.
- [5] F. Young and D. Wong. "Slicing Floorplans With Range Constraints," Proc. Intl. Symposium on Physical Design, pp.97-102, 1999.
- [6] S.N. Adya et al. "Unification of Partitioning, Placement and Floorplanning". DAC, 2004, pp. 550-557.
- [7] I. Liu, H. Chen, T. Chou, A. Aziz, D.F. Wong. "Integrated Power Supply Planning and Floorplanning", IEEE Trans. on CAD, 2004.
- [8] L. Cheng, D.F. Wong. "Floorplan Design For Multi-million Gate FPGAs", DAC, 2004, pp. 292-313.

- [9] H.Eisenmann, F.M. Johanen Genetic Global Placement and Floorplanning. DAC. San Francisko, California, 1998.
- [10] J.P. Cohoon, Distributed Genetic Algorithms for theFloorplan Design Problem, IEEE Transactions on ComputerAided Design, 10(4), 1991, 483-492.
- [11] V. Schnecke and O. Vornberger, Genetic Design of VLSI Layouts, International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995, 430-435.
- [12] Lebedev V.B. VLSI planning by method of search adaptation. Journal of TSURE news. Taganrog, 2000, №2, pp. 168-177.
- [13] Kurejchik V.M., Lebedev B.K., Lebedev O.B. Search adaptation: theory and practice. Moscow: Fizmatlit, 2006.
- [14] Practical Handbook of Genetic Algorithms. Editor I. Chambers. V.3, Washington, USA, CRC Press, 1999.
- [15] Mazumder P., Rudnick E. Genetic Algorithm For VLSI Design, Layout & Test Automation. India, Pearson Education, 2003
- [16] R.H.J.M. Otten, "Automatic floorplan design,"Proc. DAC, pp.261–267, June 1982.
- [17] C.L. Valenzuela & P.Y. Wang, VLSI Placement and Area Optimization using a Genetic Algorithm to Breed Normalized Postfix Expressions, IEEE Transactions on Evolutionary Computation, 6(4), 2002, 390-401.
- [18] Nakaya S., Koide T. and Wakabayashi S., "An adaptive genetic algorithm for VLSI floorplanning based on sequence-pair," Proc. ISCAS, pp. 65 -68, 2000.
- [19] Debarshi Chatterjee and Theodore W. Manikas. A genetic algorithm for non-slicing floorplan representation National Conference on Intelligent Systems, Hyderabad, 24-25 August 2007.
- [20] Lebedev B.K. Methodsof search adaptationin VLSI CAD. Taganrog, TSURE Publishing House, 2000.
- [21] D. F. Wong, and C.-L. Liu, "A new algorithm for floorplan design," Proc. DAC, pp. 101–107, June 1986.
- [22] MCNC Electronic and Information Technologies (Online). Available: www.mcnc.org