Module Placement on BSG-Structure and IC Layout Applications

Shigetoshi NAKATAKE†, Kunihiro FUJIYOSHI‡, Hiroshi MURATA‡, and Yoji KAJITANI †
†Department of Electrical and Electronic Engineering
Tokyo Institute of Technology
‡School of Information Science
Japan Advanced Institute of Science and Technology (JAIST)

Abstract

A new method of packing the rectangles (modules) is presented with applications to IC layout design. It is based on the bounded-sliceline grid (BSG) structure. The BSG dissects the plane into rooms associated with binary relations "right-to"and "above" such that any two rooms are uniquely in either relation. A packing is obtained through an assignment of modules on the BSG, followed by physical realization BSG-PACK. A simulated annealing searches for a good packing of all packings by changing the assignments. Experiments showed that hundreds of rectangles are easily packed in a small rectangle area (chip) with a quite good quality in area efficiency. A wide adaptability is demonstrated specific to IC layout design. Remarkable examples are: the chip is not necessarily rectangle, L-shaped modules and modules which are allowed to partially overlap each other can be handled.

1 Introduction

The placement in integrated circuit layout is, if we simply define, to locate given modules on a plane under certain *feasibility constraint* aiming at a certain *optimization target*. For example, "modules being non-overlapping each other" is a minimal feasibility constraint. "Minimizing the area of the bounding box (chip) of the modules" is a typical optimization target. The placement under the non-overlapping constraint is called the *packing*.

The packing technology is the basis of the layout. One of the major reasons of layout design automation being stumbling is due to the current inferior packing technology. Almost manual design is still prevailing for floorplan in full-custom design, printed circuit boards, and analog circuits. This paper is proposing a packing technique which will be a breakthrough to open up a way to automation. It can pack hundreds of rectangles with satisfying accuracy. It has a wide adaptability for the layout design to allow rectilinear modules (such as L-shaped modules) on a rectilinear chip or such a fuzzy request as "place the modules similarly as the diagram drawn by an expert circuit designer".

Our packing method is based on the *bounded-sliceline grid structure* or simply the *BSG*. It is a *meta-gird* which contains no physical dimension but a topological grid on a plane composed of certain orthogonal unit lines, called the *BSG-units*. As its manifest is shown in Fig.1(A), it dissects the plane into rectangular zones called the *rooms*. The BSG introduces orthogonal relations of "right-to" and "above" to the rooms unique for two of them. An assignment of given modules is to map each module to a distinct room, by which the modules inherit the relations of the rooms. Then, a process called the *BSG-PACK* outputs a packing whose bounding box has the minimum area over all the packings in which the modules satisfy the same relations.

The BSG plane is by its definition infinite. However, for the sake of computation, we restrict the plane into a finite *domain* on which assignments of modules are considered. It can be proved that any packing is the one obtained by some assignment on the domain as long as its size is not less than $n \times n$ where is n is the number of modules. This fact implies that by exhausting finite number of assignments, we can find an optimum packing.

A topologically structured region such as the BSG is called the *meta-grid*. This is not a new concept though not explicitly mentioned in general fashion so far. The *slicing structure* [1, 2] is a meta-grid which has been very popular for the sake of its simpleness in spite of its quite a loss of generality [3]. It is a "defect" grid obtained by recursive applications of "slicing a rectangle". Choices of horizontal or vertical slicing as well as its position produce a variety of structures. However, the slicing structure is very limited since trivially most of the packings are non-slicing. Many efforts [4, 5, 6] to cover this intrinsic disadvantage have been sought but not satisfactory. Hence the slicing structure is now understood that its merits are not for packing but for special purpose: Apparent advantages are in *safe ordering* of channels for detailed channel routing [7], optimizing orientations of modules [8], and finding reasonable channel sequences for global routing [9].

Another meta-grid is the *Sequence-Pair* structure which was introduced very recently by the authors [3] in which they use a 45 degree *oblique grid*. No rooms are defined but equivalently modules are assigned at the crosses. It was stressed in the paper as an es-

ICCAD '96
1063-6757/96 $5.00

sential advantage that there is an assignment which is mapped to a packing of the minimum area. This was discussed as a property of the *P-admissible solution space* over which a reasonable, and hence effective, stochastic search could be designed.

Onodera's idea [10] is very close to the meta-grid. He assigns binary relations of "right-to" and "above" uniquely to every pair of modules. Then, trivially any possible packing is surely corresponds to one of such assignments. However, as discussed in [3], there are assignments that fail to correspond to any packing.

The meta-grid BSG proposing here is not less general than the Sequence-Pair in P-admissibility and calculation speed. Its remarkable feature is the flexibility to accept such geometrical constraints as found in IC layout.

In simulated annealing implementation, it is not only capable to pack hundreds of modules in practical time with a surprising quality, but to accept various design-rules. Some of them may be such descriptions as "elements shall be placed keeping the order found in the circuit designer's diagram" or "the circuit shall be packaged within a U-shaped board with certain allowance", etc. Such design requests have been major reasons why analog layout design has been failed to be automated.

The rest of the paper is composed as follows. Sec.2 defines the BSG structure, and Sec.3 addresses how an assignment is mapped to a packing. In Sec.4, the solution space is defined as the set of all assignments on the finite domain. Sec.5 shows several hints to include essential constraints for IC layout. Experiments show the effectiveness. Sec.6 is the conclusions.

2 Bounded-Sliceline Grid Structure

A meta-grid, named the *bounded-sliceline grid* (BSG), is a topology defined on a plane. It does not include any physical dimension, but only for convenience, we use a physical image by the xy-coordinate to describe the BSG.

First, we define the unit segments. On the (x,y)-coordinate system, an open line segment $H_{i,j}$ or $V_{i,j}$ (i,j: integers) is defined by

$$\begin{aligned} H_{i,j} &= \{(x,y)|i-1<x<i+1, y=j\} \\ V_{i,j} &= \{(x,y)|x=i, j-1<y<j+1\}. \end{aligned}$$

Note that their subscripts i,j denote the centers of unit segments. The BSG is a system consisting of the set U_{BSG} of such line segments

$$\begin{aligned} U_{BSG} = \quad & \{V_{i,j}|i,j:integers,\ i+j:even\} \\ & \cup\{H_{i+j}|i,j:integers, i+j:odd\} \end{aligned}$$

See Fig.1(A). In the BSG, each $H_{i,j}$ or $V_{i,j}$ is called the (horizontal or vertical, respectively) *BSG-unit* or simply *unit*. Two vertical units $V_{i,j}$ and $V_{i',j'}$ are said *adjacent* if $|i'-i|=1$ and $|j'-j|=1$. A vertical unit $V_{i,j}$ is said *right-to* $V_{i',j'}$ if $i'=i-1$ and $|j'-j|=1$. The relation "right-to" is extended transitively: If a vertical unit V_a is right-to another vertical unit V_b and

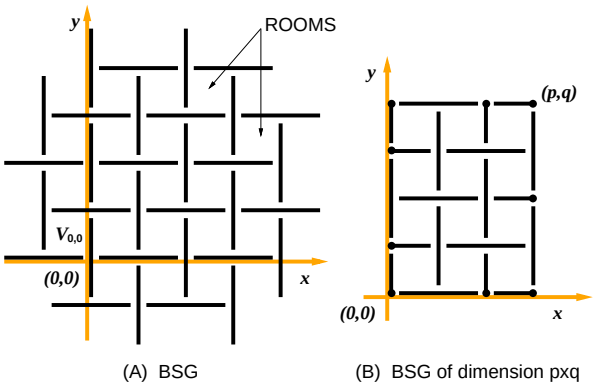


Fig.1: BSG and a Domain $BSG_{p\times q}$

V_b is right-to V_c, then V_a is right-to V_c. For horizontal units, analogous relations *"above"* and *"adjacent"* are defined.

A rectangular space surrounded by adjacent pairs of vertical- and horizontal units is called the *room*.

By definition, the BSG is an infinite grid. But as a work-sheet, it is convenient to bound the grid within a finite grid $BSG_{p\times q}$ (p,q: positive integers) of p rows and q columns of rooms whose bottom-left corner is the origin $(0,0)$ and the above-right corner is (p,q). We call it the *domain* of size $p\times q$. For compactness, portions of units jutting outside the domain are cut off. Therefore, as shown in Fig.1(B), some of peripheral units are cut into a half, e.g. an open segment $V_{0,0}$ becomes a half-open segment $\{(x,y)|0\le x<1, y=0\}$.

In general, a directed graph is denoted as $G(V,E)$ where $(u,u')\in E$, $u,u'\in V$, represents a directed edge from u to u'. Here, two digraphs, *horizontal unit adjacency graph* $G_h(V_h,E_h)$ and *vertical unit adjacency graph* $G_v(V_v,E_v)$, are defined to represent the orthogonal binary relations "right-to" and "above".

Given a domain $BSG_{p\times q}$, $V_h = \{s_h,t_h\}\cup\{u_{i,j}\}$ where $u_{i,j}$ corresponds to the unit $H_{i,j}$ ($i+j$: odd). Edges are defined as follows. s_v is a source connected to all the vertices corresponding to the bottom units, *i.e.* $H_{1,0}, H_{3,0}, \cdots, H_{2i+1,0}$ where $i=\lfloor(p-1)/2\rfloor$. t_h is a sink connected from all the vertices corresponding to the top units which are, for example of the case q: even, $H_{1,q}, H_{3,q}, \cdots, H_{2i+1,q}$ where $i=\lfloor(p-1)/2\rfloor$. The other edge $(u_{i,j}, u_{i',j'})$ exists if and only if $H_{i',j'}$ is above and adjacent to $H_{i,j}$.

The vertical unit adjacency graph $G_v(V_v,E_v)$ are similarly defined. Examples shown in Fig.2 will make these definitions clear.

If we draw, as in this figure, these graphs over $BSG_{p\times q}$ putting the vertices on the centers of units, each room is crossed exactly by one edge in each of G_h and G_v. By this relation, an edge and a room are conveniently referred to by the other in such a fashion as "an edge that crosses room r", or "a room which edge e crosses".

The following lemma is easily understood from the observation of $BSG_{p\times q}$. See Fig.3.

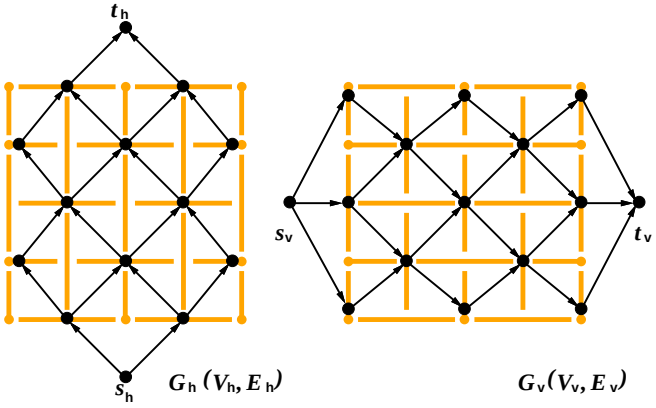


Fig.2: Horizontal Unit and Vertical Unit Adjacency Graphs

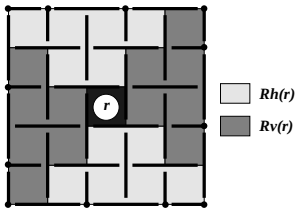


Fig.3: $R_v(r)$ and $R_h(r)$

Lemma 1 (room relations):

Let r be any room and e_r^v and e_r^h be edges crossing r in G_v and G_h, respectively. Let $R_v(r)$ be the set of rooms whose crossing edges are contained in a directed path that contains e_r^v in G_v. Similarly, let $R_h(r)$ be the rooms whose crossing edges are contained in a directed path that contains e_r^h in G_h. Then, it holds

$$R_v(r) \cap R_h(r) = \{r\} \text{ and}$$
$$R_v(r) \cup R_h(r) = \text{all-the-rooms.}$$

Thus, room r' ($\neq r$) is in $R_v(r)$ or in $R_h(r)$, but not in both. In the case of the former, r and r' are in "right-to" relation, else they are in "above" relation. Accordingly, two rooms are uniquely defined in either of relations "right-to" and "above".

3 From an Assignment to a Packing

Suppose we are given an input. i.e. a set of modules M, where $|M| = n$. Assuming that $p \times q \geq n$, an *assignment* of M is a one-to-one mapping of modules into the rooms of $\mathrm{BSG}_{p\times q}$. A room to which no module is assigned is *empty*.

Weighting of unit adjacency graphs $G_v(V_v, E_v)$ and $G_h(V_h, E_h)$ is to associate each edge e with a real number $w(e)$ by the following formula:

If $e \in E_h$ and e crosses a non-empty room,
$w(e)$ =height of the module assigned there.

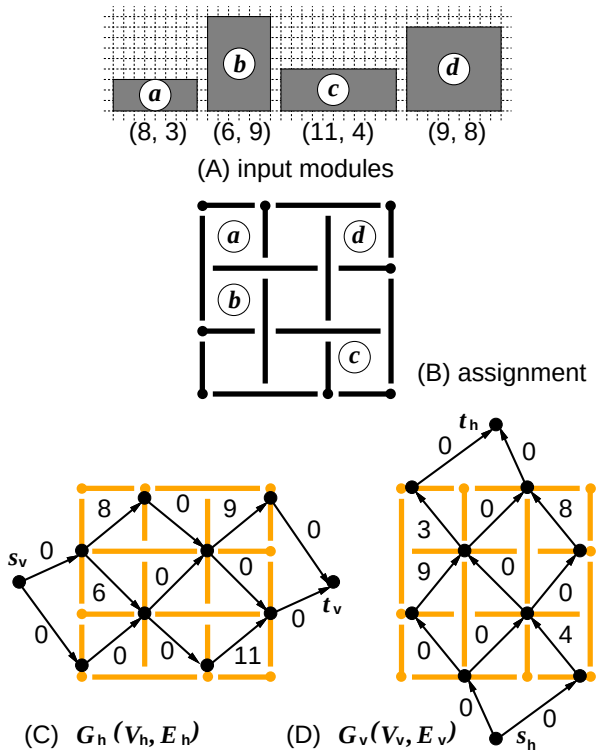


Fig.4: Given Modules, Assignment, and Edge-Weighting of the Unit Adjacency Graphs

If $e \in E_v$ and e crosses a non-empty room,
$w(e)$ = width of the module assigned there.

Otherwise, that is, if e is either to cross an empty room or has its endvertex on the source or sink,
$w(e) = 0$.

An example up to this stage is shown in Fig.4.

Let $G_h(V_h, E_h)$ be the horizontal unit adjacency graph thus edge-weighted. For each vertex $u \in V_h$, $l_h(u)$ denotes the length of the longest path from the source s_h to u. Similarly in G_v, $l_v(u)$ denotes the longest path length from s_v to $u \in V_v$.

The procedure to find $l_h(u)$ and $l_v(u)$ for every $u \in V_h$ and $u \in V_v$ is a well-known technique which we refer to **procedure: LONGEST-PATH LENGTH (G)** where G is the input, i.e., $G_h(V_h, E_h)$ or $G_v(V_v, E_v)$. It works in linear time of the number of edges when the input G is a directed acyclic graph (DAG). The total number of edges of the unit adjacency graphs is between $2(pq + p + q)$ and $2(pq + p + q) - 4$, we have the following fact.

Lemma 2 (time complexity):

The time complexity of **LONGEST-PATH LENGTH (G)**, where G is $G_h(V_h, E_h)$ or $G_v(V_v, E_v)$, is $O(pq)$.

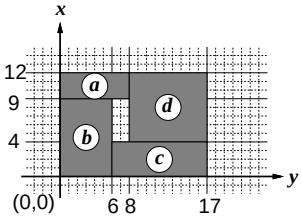


Fig.5: (Continued) Packing by BSG-PACK

The purpose of computing the longest path length in the unit adjacency graphs is to determine the positions of modules.

Theorem 1 (BSG-placement):

Given an assignment of M to $\mathrm{BSG}_{p\times q}$, the placement by the following procedure provides a packing.

procedure: BSG-PACK (assignment)

Let m be a module assigned to a room whose boundary left vertical unit is V_m and bottom horizontal unit is H_m. Then, place m such that its left bottom is at $(l_v(u_{Vm}), l_h(u_{Hm}))$ where u_{Vm} and u_{Hm} are the vertices corresponding to units V_m and H_m in the vertical unit and horizontal unit adjacency graphs, respectively.

We have only to prove that no two modules overlap. This is not difficult if we notice that any two rooms in right-to (above) relation keep the relation in the output of the procedure since those corresponding edges are on a directed path.

The following fact is direct:

Corollary 1 (chip area): The area (of the minimum bounding box, chip) of the packing by **BSG-PACK (assignment)** is $(l_v(t_v) \times l_h(t_h))$.

In Fig.5, the output of the procedure for the assignment in Fig.4 is shown.

Thus, the procedure **BSG-PACK** introduces the physical dimension to the meta-grid.

The output of **BSG-PACK** may allow further one-dimensional compaction to reduce the area, if it were allowed for modules to penetrate their peripheral BSG-units. However, still we stop at this half-way compaction because, as will be discussed later, it is sufficient to guarantee the possibility of finding an optimum solution in searching the solution space.

Note that the horizontal and vertical compactions are independent, i.e., whichever is applied first the result is unique.

4 Solution Space and P-Admissibility

The solution space $\mathcal{S}_{p\times q}$ consists of all possible assignments on $\mathrm{BSG}_{p\times q}$. Since an assignment is mapped to a unique packing by the BSG-PACK, $\mathcal{S}_{p\times q}$ is equivalently defined as the set of packings by BSG-PACK, though more than one assignments may be mapped to an identical packing.

Let us introduce an evaluating function of each packing in $S_{p\times q}$. Any function is valid as long as it evaluates the "quality" of the resultant packing. Here, let us take a simple one, the minimum area of the chip, which we call simply the *area* of the packing, taking it the smaller the better. It is $l(t_h) \times l(t_v)$ by Corollary 1. A solution whose area is minimum is the *optimum packing*.

Consider to search the space looking for a better packing. The facts will help the search that the space is finite, evaluation of each solution is quick (Corollary 1), and every assignment maps to a packing. However, the effort will be most encouraged if it is guaranteed that an optimum solution is included in the space. It depends on the size of domain $\mathrm{BSG}_{p\times q}$.

Theorem 2 (optimum solution):

The solution space $\mathcal{S}_{p\times q}$ contains one of the optimum solutions if $p, q \geq n$. Furthermore, if p or q is less than n, there is an input M which does not have any assignment to lead the optimum packing of M.

(Our proof is so complicated out of readers' interest, so omitted.)

Thus, if the domain is $\mathrm{BSG}_{n\times n}$ or larger, the solution space $\mathcal{S}_{p\times q}$ is the one called *P-admissible* [3]. In the paper, it was discussed that the P-admissibility is the minimum requirement for the space on which solutions are searched. For the packing problem, constructing a P-admissible solution space is not trivial. One based on the sequence-pair [3] was the first one. Our BSG-based solution space is the second if we take the domain large enough.

Expecting one of the optimum solutions, let $p = q = n$. The variety of assignments is "n distinct objects from n^2", i.e. $n^2!/(n^2-n)!$. This number easily explodes: for small ones as $n = 10$, $n = 15$, and $n = 20$, it is about 10^{20}, $3 \cdot 10^{35}$, and 10^{52}, respectively.

The packing problem was proved [3] to be NP-hard. Exhausting the solutions of $\mathcal{S}_{n\times n}$ will not end in practical time whatever the computation resources are. So letting us satisfy ourselves with some approximations, we took a stochastic way: a very standard simulated annealing was implemented. It is simply described as follows:

Sketch of the Simulated Annealing: Start with any assignment, get the corresponding packing by BSG-PACK, calculate the area, store the packing as a current solution. Then, (*) *change* the assignment, get the corresponding packing, replace the current solution with this if the area is decreased or "sometimes"

even if the area is increased according to the *probability*, change the assignment, *lower* the temperature, and return to (*).

In the procedure, change of the assignment is executed by interchanging (swapping) the contents of randomly chosen two rooms, and the probability is controlled by the parameter *temperature*. It is lowered following the prescribed schedule until to freeze, that is, to shift to the quenching process. The procedure stops then.

There are ideas to improve the approximation with limited computation resources, for example, in restricting the space, in generating the next solution, in scheduling to lower the temperature, in multi-solution annealing. For our BSG-based packing, there is a specific way which is to restrict the space, that is, to use $\mathrm{BSG}_{r\times r}$ instead of $\mathrm{BSG}_{n\times n}$ taking r far smaller than n.

Actually, the assignment on $\mathrm{BSG}_{n\times n}$ is very much time consuming for BSG-PACK: $O(n^2)$ is large for one cycle time in simulated annealing. For example, for $n = 500$, if we let $r = 50$, a cycle time reduces to $1/100$, in other words, the simulated annealing can search the solutions 100 times more within the same time. The trade-off of this speedup is the loss of guarantee of containment of optimum solutions. It is a difficult problem to determine what size of the BSG is practically optimum. Some compromise is discussed through experiments in Appendix. Its conclusion is optimistic that a standard simulated annealing can get a fairly good packing unless r is close to $\sqrt{n}$. Packings of 146 modules by various size BSG's are shown in Fig.6. The achieved area for $\mathrm{BSG}_{13\times 13}$ is significantly larger than the others.

5 Basic Techniques in BSG-Based Packing

This section is devoted to show how the BSG-based packing works so adaptive to the complicated layout designs taking examples in analog and PCB circuit layouts.

5.1 Rectilinear Chip

Packings into a general rectilinear chip were successful by the following strategy. Preassign appropriate dummy modules with appropriate sizes on the specified area of the BSG (See Fig.7). In simulated annealing, the assignment of the dummy modules will not be changed. Then, other modules keep away there and the resultant packing will take a nearly desired shape. In Fig.8, 200 modules are packed in three ways of different chips (in letters C, A, or D). Another example is shown in Fig.11.

However, since the peripheral constraint include some physical dimension, the constraint is not always met under the strategy to minimize the area. To solve this mismatch is one of future works. (See Conclusions.)

5.2 Rectilinear Modules

It is often in IC layout to handle non-rectangular modules, typically, L-shaped modules [11, 12, 13]. In the BSG-based layout, it does not need extra computation to allow such specifications. An L-shaped module is embedded by the technique shown in Fig.9: Cut the module into two rectangles so that the cut-line and even-line are defined as in the figure. Assign two pieces to such a pair of rooms that share the units as these cut-line and even-line. There are several ways of such assignments. In the figure, the placements in the left and right show one and its rotated placement.

This is the simplest example of successful non-rectangular embeddings. It is not yet clear what kinds of shapes can be thus embedded.

For example, see Fig.10. 40 modules, 10 of them are L-shaped, are packed so nicely in 5 minutes, where rotation and inversion were considered.

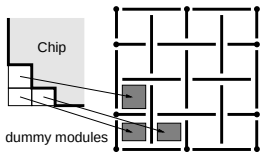


Fig.7: Rectilinear Chip Packing Using Dummy Modules

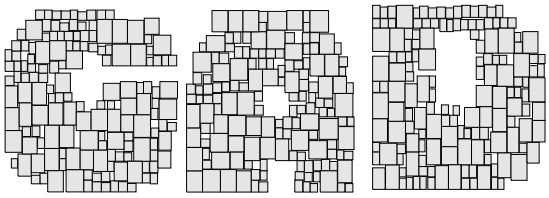


Fig.8: Packings into Chips Shaped "C", "A" or "D"

5.3 Rotation, Inversion, and Soft Modules

It is hoped in practice that rotation by multiple of 90 degrees or inversion of each module can be handled. Our idea is to prepare a list of templates. For each general rectangle module with pins, 8 different modules are registered. In simulated annealing we choose one of them to represent the module in consideration. The solution space will expand 8^n times. It looks tremendous, but experiments showed that this does not degrade the quality.

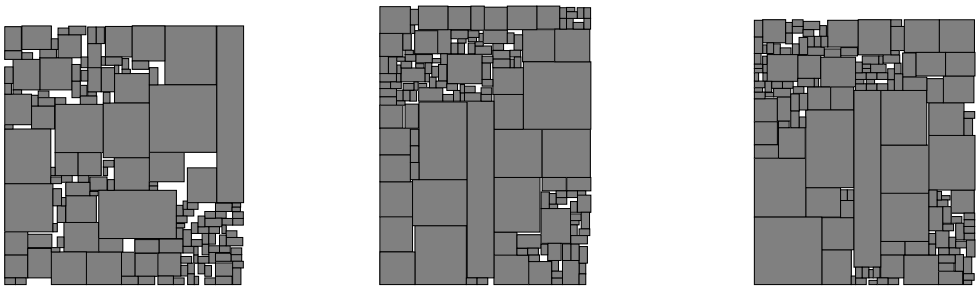
For soft modules whose aspect ratio is variable, a similar idea applies. Preparing a certain number of templates to represent a module, assignment chooses one of them each time.

6 Techniques Specific to PCB and Analog IC Layout

Several essential techniques are briefly described.

6.1 Wire Density Driven Placement

In printed circuit board (PCB) layout design, the key issue in placement is to secure the enough area for wiring. For the purpose we define the *wire density of a room* in BSG as the count of the bounding boxes of the terminals of nets. (A similar idea is found in [14].)



(a) by 13×13 (area: 861672) (b) by 20×20 (area: 819520) (c) by 25×25 (area: 814936)

Fig.6: 146 Module Packings by Various Size BSG's

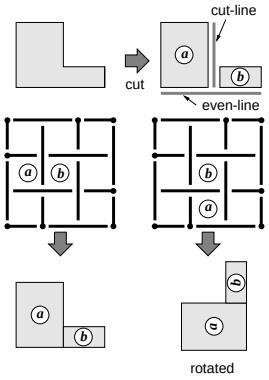


Fig.9: Procedure of Embedding L-shaped Modules

Fig.10: Packing of Modules Including L-shaped Modules

If a module is assigned a room, then we overestimate the size of the module to certain extent according to the wire density of the room. This technique was shown very successful in practical applications as shown in Fig.11, where we applied the rectilinear chip placement technique for the complex shaped board using dummy modules as shown in the left of the figure.

6.2 Circuit Theoretical Placement

In analog (leaf cell) design, it is believed that the circuit performs good when elements are laid out faithfully to the diagram described by the circuit designer [15]. Furthermore, it is often the designer specifies ad hoc requests: A typical request is "specific modules should be placed close each other". The BSG-based packing can meet such requests simply by restricting the assignments of modules on the BSG.

6.3 Multi-Border Module Placement

One of difficult cases which has been blocking automated layout design is in the existence of *multi-border modules*. A transistor consists of diffusion and isolation layers, while a resistor does of the polysilicon layer. If the chip consists of two layers and diffusion and polysilicon are laid out on the same layer, isolation and polysilicon are allowed to overlap each other but not diffusion and polysilicon. If we see the projection, the transistor looks a rectangle with double borders. In order to make module close as possible (for area reduction as well as performance), we should make use of this allowance. The BSG-packing formulates the situation as follows.

Let 'diffusion' and 'polysilicon' correspond to the 1st layer, and 'isolation' to the 2nd layer. Prepare two BSG's and correspond each BSG to each layer. Let a double-border module be m and assign m to both rooms of the same address of these BSG's. (See Fig.12.) Let two left vertical BSG-units of the assigned room be V_1 and V_2. Then move m from right to left until the left boundary of the 1st layer of m hits V_1 or that of the 2nd layer m does V_2. Next, let two right vertical BSG-units of the assigned room be V_1' and V_2'. Move V_1' and V_2' to the right boundaries of the 1st and 2nd layer of m, respectively. This is how a module is determined its x-coordinate. The y-coordinate is determined similarly.

Every module is fixed its x-coordinate and y-coordinate independently, one by one from the modules assigned in the leftmost rooms and bottom rooms, respectively.

One example is shown in Fig.13. where diffusions do not overlap but isolations do to lead a layout of reduced area.

Industrial samples were designed by our packing method considering inversion and multi-border techniques. In Tab.1 the comparison with manual designs is shown. Our result is definitely superior with respect to the area. TEST1 and TEST4 are illustrated in Fig. 14.

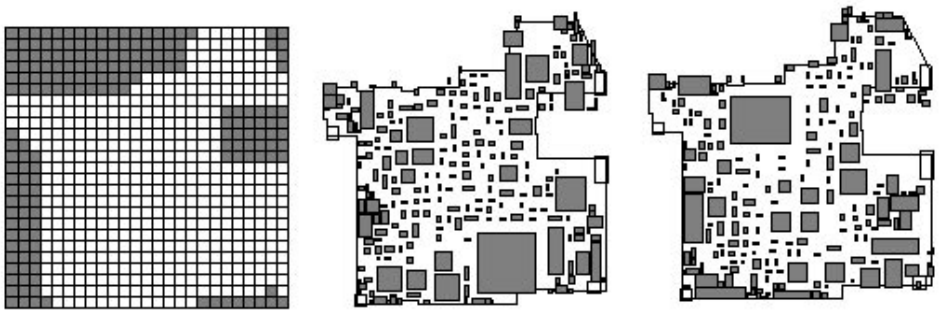


Fig.11: An Industrial Example PCB by Wire Density Driven Placement: Front and Back Boards for Digital Video Movie with 468 Components and 1389 Nets.

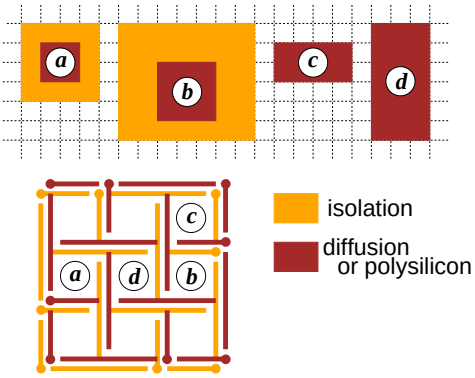


Fig.12: Packing of Multi-border Modules

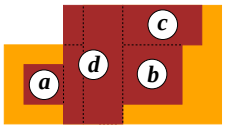


Fig.13: (Continued) Packing of Multi-border Modules

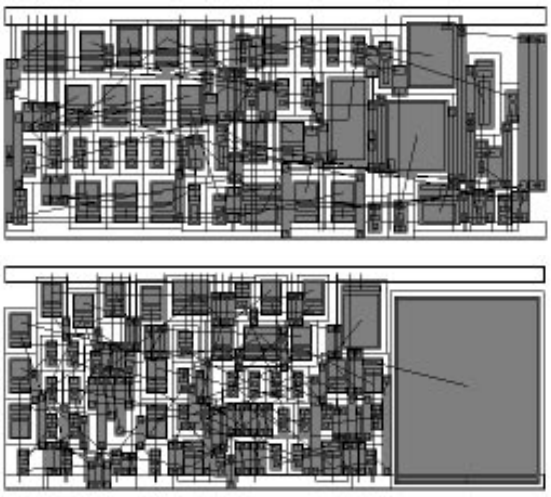


Fig.14: Automated Layout Designs of Analog Circuits TEST1 and TEST4

7 Conclusions

We introduced a new meta-grid, the Bounded-Sliceline Grid. It provides a P-admissible solution space if the size of BSG is not less than $n \times n$. A heuristics using simulated annealing was proposed which stochastically searches the space. In spite of hugeness of the space, experiments showed that it performs very well, practically enough.

This heuristics revealed great adaptability to the complex placements such as analog IC's and PCB's. Techniques were developed to treat rectilinear chips, rectilinear modules, multi-border modules, and to meet fuzzy circuit theoretical specifications.

Further researches are to get a more reasonable rule-of-thumb on triple trade-off among quality, computation time, and domain size. (See Appendix.) And for IC design automation, they are to develop techniques: to reduce computation time including soft modules, to estimate precise area and length of wiring, to meet the peripheral physical dimension of the chip, and to evaluate the "quality" of complicated various design specifications on the resultant chip.

circuits	size of base(W x H)		time (sec)
	manual	BSG	BSG
TEST1	700 x 250	594 x 255 (−13%)	389.49
TEST2	645 x 290	580 x 299 (−7%)	467.84
TEST3	685 x 290	606 x 293 (−11%)	354.08
TEST4	750 x 290	701 x 288 (−7%)	484.23

Tab.1: Four Analog IC layouts including about 100 modules

Acknowledgments

The authors are grateful to Mr. Eiichi Kaneko and his group, Apollo Co., for supplying data for industrial PCBs. They also appreciate Dr. Masahiro Kawakita

and his group, Toshiba Co., for their sincere discussions in developing analog IC layouts. Finally it must be mentioned that this research is partly supported financially by CAD21 at Japan Advanced Institute of Science and Technology.

References

[1] R.H.J.M.Otten, "Automatic Floorplan Design", Proc. 19th ACM/IEEE Design Automation Conf. pp.261-267, 1982.

[2] D.F.Wong and C.L.Liu, "A New Algorithm for Floorplan Designs", Proc. 23rd ACM/IEEE Design Automation Conf. pp.101-107, 1986.

[3] H.Murata, K.Fujiyoshi, S.Nakatake and Y.Kajitani, "Rectangle-Packing-Based Module Placement", Proc. International Conf. on CAD pp.472-479, 1995.

[4] W.M.Dai and E.Kuh, "Simultaneous Floorplanning and Global Routing for Hierarchical Building Block Layout", IEEE Trans. on CAD Vol.6 No.5 pp.828-837, 1987.

[5] T.C.Wang and D.F.Wong, "An Optimal Algorithm for Floorplan Area Optimization", Proc. 27th ACM/IEEE Design Automation Conf. pp.180-186, 1990.

[6] T.C.Wang and D.F.Wong, "A Graph Theoretic Technique to Speed up Floorplan Area Optimization", Proc. 29th ACM/IEEE Design Automation Conf. pp.62-68, 1992.

[7] Y. Kajitani, "Order of Channels for Safe Routing and Optimal Compaction of Routing Area", IEEE Trans. on CAD Vol.2 No.4 pp.293-300, 1983.

[8] L.Stockmeyer, "Optimal Orientations of Cells in Slicing Floorplan Designs", Information and Control Vol.59 pp.91-101, 1983.

[9] H.Matsuda, S.Nakatake and Y.Kajitani, "Optimum Slicing-Structure Floorplanning with Routing Area Included", IEICE Technical Report VLD94-109 Vol.94 No.531 pp.9-14, 1995.

[10] H.Onodera, Y.Taniguchi and K.Tamaru, "Branch-and-Bound Placement for Building Block Layout", Proc. 28th ACM/IEEE Design Automation Conf. pp.433-439, 1991.

[11] J.A.Hudson, R.C.Peters, "Module Positions Algorithms for Rectilinear Macrocell Assembles", Proc. Design Automation Conf. pp.672-675, 1984.

[12] M.C.Chi, "An Automatic Rectilinear Partitioning Procedure for Standard Cells", Proc. Design Automation Conf. pp.50-55, 1987.

[13] T.Lee, "Bounded 2D Contour Searching Algorithm for Floorplan Design with Arbitrarily Shaped Rectilinear and Soft Modules", Proc. Design Automation Conf. pp.525-530, 1993.

[14] S.Nakatake, Y.Kajitani, "Channel-Driven Global Routing with Consistent Placement", Proc. International Conf. on CAD pp.350-355, 1994.

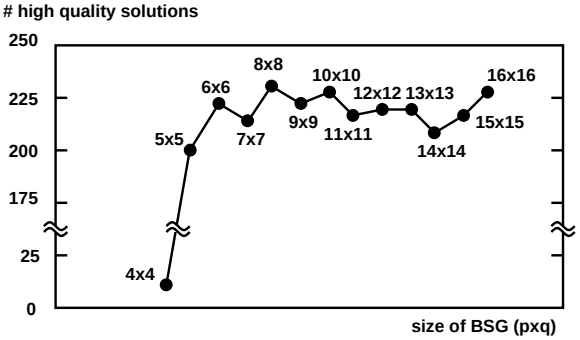
[15] Y.Shiraishi, M.Kimura, K.Kobayashi, T.Hino, M.Seriuchi, and M.Kusaoke, "A High-Packing Density Module Generator for Bipolar Analog LSIs", Proc. International Conf. on CAD pp.194-197, 1990.

Appendix: Features of the Solution Space and SA Strategies

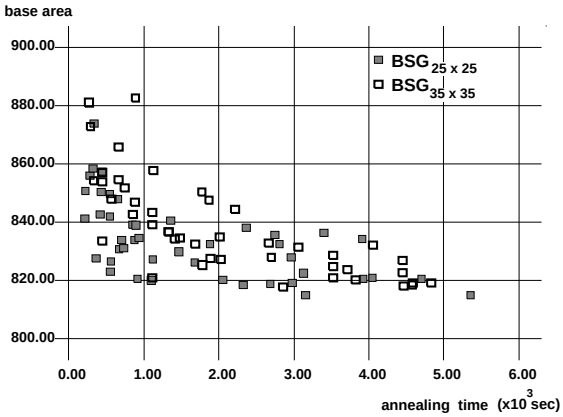
To explore the feature of our heuristics with respect to the triple trade-off in "quality", "domain size", and "computation time" we tried the following experiments to lead a "rule-of-thumb".

M is a set of 16 modules of random sizes. Consider various domains $\mathrm{BSG}_{4\times4}$, $\mathrm{BSG}_{5\times5}$, $\cdots$, $\mathrm{BSG}_{16\times16}$.

First, to see the distribution of good solutions, a random search was executed: To each BSG, take 50 million random assignments, and classify the resultant packings into 7 classes according to their quality (smallness of chips). The packings in Class 1 and Class 2, whose area is within 120 % of the sum of module sizes,



(a) Quality versus Domain Size in Random Search in case of 16 Module Packing



(b) Quality versus Time in SA in case of 146 Module Packing

Fig.15: Trade-offs of Quality, Domain Size, and Computation Time

are considered to have the quality enough for practical use. (See Fig.15 (a))

Then, an observation tells us that too small a domain such as $r = 4$ is not recommended. But once the percentage of good packings jumps up at $r = 5, 6, 7, 8, 9, ..$, significant improvement is observed no more. We might conclude that if you have a result of fairly good quality for some r, you should not be worried about the degradation caused by reducing the domain size; you will not get significant improvement by expanding the domain.

Second, our experiment concerns with the convergence of the quality in simulated annealing. Theoretically we can say that: As the size of the BSG domain is large, (1) the possibility of the solution space including an optimum becomes high, and (2) the solution space becomes huge. Therefore it is not a simple matter what size is good for available computing resources. We experimented an example of 146 modules for quality versus time. Among many data, two are shown in Fig.15 (b). In this case, time of 15 minutes is sufficient to get practically good solution if you use $\mathrm{BSG}_{25\times25}$.