Rectangle-Packing-Based Module Placement *

Hiroshi Murata †, Kunihiro Fujiyoshi †,
Shigetoshi Nakatake †, and Yoji Kajitani ‡

†School of Information Science, Japan Advanced Institute of Science and Technology
Tatsunokuchi, Ishikawa 923-12, Japan

‡Department of Electrical and Electronic Engineering, Tokyo Institute of Technology
Meguro-ku, Tokyo 152, Japan

abstract

The first and the most critical stage in VLSI layout design is the placement, the background of which is the rectangle packing problem: Given many rectangular modules of arbitrary size, place them without overlapping on a layer in the smallest bounding rectangle. Since the variety of the packing is infinite (two-dimensionally continuous) many, the key issue for successful optimization is in the introduction of a P-admissible solution space, which is a finite set of solutions at least one of which is optimal. This paper proposes such a solution space where each packing is represented by a pair of module name sequences. Searching this space by simulated annealing, hundreds of modules could be successfully packed as demonstrated. Combining a conventional wiring method, the biggest MCNC benchmark ami49 is challenged.

1 Introduction

Layout in physical design of VLSI is, simply to say, to pack all the circuit elements in a chip without violating the design rules, so that the circuit performs well and the production yield is heigh. So much the variety of targets in different stages, the problem defined as follows is the base of all of them.

Rectangle Packing Problem: RP
Let $\mathcal{M}$ be a set of m rectangular modules whose height and width are given in real numbers. (Orientation is fixed.) A packing of $\mathcal{M}$ is a non-overlapping placement of the modules. The minimum bounding rectangle of a packing is called the chip. Find a packing of $\mathcal{M}$ in a chip of the minimum area.

A packing of six modules is shown in Fig.1.

It has been known that the decision problem [**RP(H,W):** Given a base-rectangle with height H and width W, determine if a given set $\mathcal{M}$ of modules can be packed] is NP-complete[1]. The decision version of our packing problem is [**RP(A):** Given a base-rectangle with area A, determine if a given set $\mathcal{M}$ of modules can be packed].

This problem is shown to be NP-complete since: if we let $A = (H+W)(2H+W)$ and the set of modules be $\mathcal{M} \cup \{H \times H, (H+W) \times (H+W)\}$, **RP(H,W)** is polynomially reduced to **RP(A)**. Therefore, our problem **RP** is harder than **RP(A)**, so NP-hard.

Since the height and width of modules are continuous real numbers, **RP** is not simply a combinatorial optimization problem. Hence there have been several numerical approaches[2, 3].

*This work was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science, and Culture of Japan (05452209) and Research Body CAD21 at JAIST.

An alternative approach is "combinatorial search". Define a solution space which is a set of codes. Each code represents a construction of placement. A code is said to be *feasible* if the construction is consistent, i.e. there exists a packing corresponding to the code. The evaluation of a code is the evaluation of the packing (the area of the chip). Search a feasible code which yields the best packing.

If a trade-off to the computation time is imposed, the heuristics will stop the search on the way. Being effective this search, the minimum requirement of the solution space is

(1) The solution space is finite,

(2) Every solution is feasible,

(3) Evaluation for each code is possible in polynomial time and so is the realization of the corresponding packing,

(4) The packing corresponding to the best evaluated code in the space coincides with an optimal solution of **RP**.

The solution space that satisfies the above four requirements is called *P-admissible.*

The reasons for (1),(3) and (4) are obvious. That for (2) is: most heuristics pick up one solution after another along the neighboring structure defined on the space[4], consulting with the difference of evaluations (gain) to the previous solution. Therefore, if infeasible solutions are included, the continuity will be destroyed and the convergence to a feasible solution is not guaranteed.

A practically known solution space is the one derived from *slicing floorplan* proposed by Otten[5] and others. Because it satisfies (1), (2) and (3), and also because it has "safe routing" merit for channel routers[6], several optimization heuristics are applied for the space, and one of the most successful approaches uses simulated annealing[7]. However, since the optimal solution can be non-slicing, it lacks (4). This fact discourage us to start searching the best in the space. Efforts have been paid to add non-slicing structures to the space[8, 9], but they are not successful to cover the defect.

On the other hand, Onodera et.al.[10] uses a solution space by assigning one out of four relations, "left to" , "right to", "above", "below", to every pair of modules. This space satisfies (4) since any packing satisfies a combination of the relations. But there are many infeasible codes such as; module a is left to module b, b is left to module c and c is left-to a. As a consequence, the space does not admit such a heuristics as simulated annealing. In their paper, an exhaustive search with a branch-and-bound technique is applied to find an exactly optimal solution, but the size of tractable problems is limited up to six modules. Thus these two are not P-admissible.

This paper provides a P-admissible solution space, in which each code is a pair of module name sequences. Searching this space, hundreds of modules have become able to be packed very efficiently, almost optimally at a look as in Fig.7 and in Fig.8.

Utilizing this solution space of **RP** for VLSI layout design, the evaluation of a packing has to be modified to consider wires. Many formulae have been proposed for this purpose[7, 10], and we follow [10]. The largest MCNC building-block benchmark is successfully placed by simulated annealing in about 30 minutes(Fig.9).

This paper is organized as follows. In Section 2, a mapping from a given packing to a pair of module name sequences is given, to show that an optimal solution is included. Section 3 provides a procedure of an inverse mapping from a sequence pair to a packing. Section 4 demonstrates how the space can be utilized in placement problems. Section 5 is for concluding remarks.

2 From a Packing to a Sequence-Pair

Let Π be a packing on chip C. See Fig.1 for an example. We describe a procedure called **Gridding**, which encodes Π to a *sequence-pair*, which is an ordered pair of module name sequences.

2.1 Gridding

For each module x, we draw lines using a pebble p (which is regarded as the nib of a pen). The pebble p is initially placed at the center of x and moved rightward. It turns its direction up, right, up, right, $\cdots$ whenever it encounters a object of three kinds, which are (i) the boundary of a module, (ii) the boundary of the chip, and

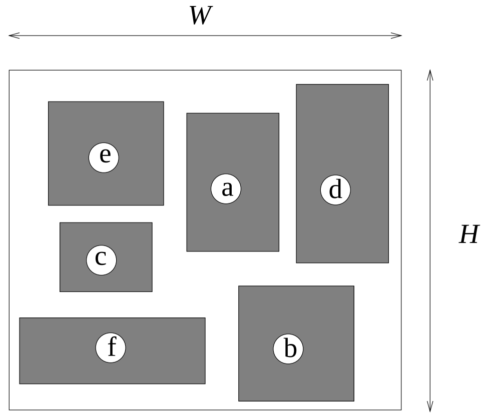


Fig.1: A packing in a chip of area $H \times W$

(iii) a previously drawn line. p stops when it reaches the upper right corner of the chip. The drawn line called the *right-up step-line* of the module x. Similarly, *left-down step-line* of x is drawn. The union of these two step-lines is called the *positive step-line* of x (since it tends to go inside the 1st and 3rd quadrants). For every module, one positive step-line is uniquely defined. They are reffered to be the corresponding module names. For example, Fig.2 is obtained if we draw positive step-line for each module in Fig.?? in the order a, b, c, d, e, f.

From construction, no two positive step-lines cross each other. It implies that the modules are linearly ordered. Here we order the positive step-lines from upper left. Let Γ_+ be the module name sequence in this order. From Fig.2, "$\Gamma_+ = ecadfb$" is obtained.

Negative step-lines are drawn similarly to positive step-lines. The difference is, the negative step-line is the union of *up-left step-line* and *down-right stepline*, whose direction changing policy are "up,left,up,left,$\cdots$", and "down,right,down,right,$\cdots$", respectively. An example is shown in Fig.3. We order the negative step-lines from lower left. Let Γ_- be the module name sequence in this order. In the example case of Fig.3, "$\Gamma_- = fcbead$" is obtained.

The ordered pair (Γ_+, Γ_-) is the resultant sequence-pair of the procedure **Gridding**.

2.2 Geometrical Information of Sequence-Pair

Let (Γ_+, Γ_-) be the sequence-pair produced by **Gridding** for the packing Π. For a module x, any other module x' is uniquely one of four cases, x' is after/before x in Γ_+/Γ_-. Let us define four classes, accordingly.

$$\begin{aligned}
\mathcal{M}^{\mathrm{aa}}(x) = &\ \{x' \mid x' \text{ is after } x \text{ in both } \Gamma_+ \text{ and } \Gamma_-\}, \\
\mathcal{M}^{\mathrm{bb}}(x) = &\ \{x' \mid x' \text{ is before } x \text{ in both } \Gamma_+ \text{ and } \Gamma_-\}, \\
\mathcal{M}^{\mathrm{ba}}(x) = &\ \{x' \mid x' \text{ is before } x \text{ in } \Gamma_+ \text{ and after } x \text{ in } \Gamma_-\}, \\
\mathcal{M}^{\mathrm{ab}}(x) = &\ \{x' \mid x' \text{ is after } x \text{ in } \Gamma_+ \text{ and before } x \text{ in } \Gamma_-\}.
\end{aligned}$$

For example, with respect to $(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$, four subsets for module c are: $\mathcal{M}^{\mathrm{aa}}(c) = \{a, b, d\}$, $\mathcal{M}^{\mathrm{bb}}(b) = \emptyset$, $\mathcal{M}^{\mathrm{ba}}(b) = \{e\}$, and $\mathcal{M}^{\mathrm{ab}}(c) = \{f\}$.

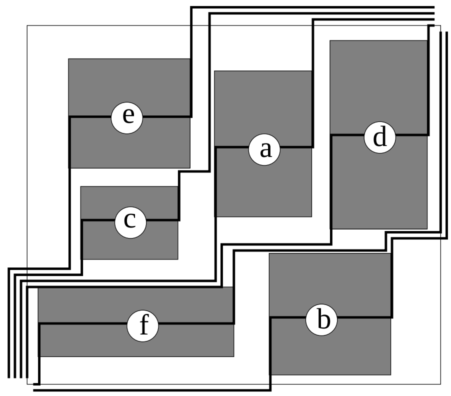


Fig.2: Positive step-lines

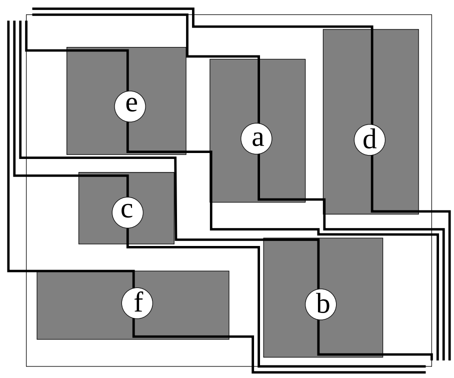


Fig.3: Negative step-lines

Any module other than x belongs to a unique subset, and it is trivial that two modules are in a dual relation through $x \leftrightarrow x'$, and $a \leftrightarrow b$ as:

$$
\begin{aligned}
x' \in \mathcal{M}^{\mathrm{aa}}(x) \quad &\Leftrightarrow \quad x \in \mathcal{M}^{\mathrm{bb}}(x') \\
x' \in \mathcal{M}^{\mathrm{ba}}(x) \quad &\Leftrightarrow \quad x \in \mathcal{M}^{\mathrm{ab}}(x')
\end{aligned}
$$

In a packing, if the left side of module x is right to the right side of module x', x is said to be *right to* x'. Similarly, *left to*, *above*, *below* relations between two modules are defined. These notations follow [10].

Theorem 1 :

Let (Γ_+, Γ_-) be the sequence-pair produced by **Gridding** for the packing Π. If $x' \in \mathcal{M}^{\mathrm{aa}}(x)$, then x' is right to x in Π.

The claim holds replacing the pair of words ("$\mathcal{M}^{\mathrm{aa}}$" and "right to") with any of ("$\mathcal{M}^{\mathrm{bb}}$" and "left to"), ("$\mathcal{M}^{\mathrm{ba}}$" and "above"), and ("$\mathcal{M}^{\mathrm{ab}}$" and "below").

Before the proof, an example is shown. From Fig.2 and Fig.3, modules a, b, d are in $\mathcal{M}^{\mathrm{aa}}(c)$, and in Fig.1, they are right to c in Π.

Proof: We sketch the proof taking an example of Fig.2 and Fig.3. Pick arbitrary two modules, c and b. The step-lines of c divide the chip into four regions. Among them, the region surrounded by the right-up step-line of c, down-right step-line of c, and the right side of the chip is called the *right-cone* of c. Similarly, the *left-*, *above-*, and *below-cone* denote other three regions.

Suppose b is in $\mathcal{M}^{\mathrm{aa}}(c)$. This implies that the positive step-line of b is in the union of the right-cone and below-cone of c. Also it is implied that the negative step-line of b is in the union of the right-cone and above-cone of c. The cross point of the positive and negative step-lines of b is in their intersection, that is, the right-cone of c. Then the module b is in the right-cone of c. All the modules in the right-cone of c must be right to module c by definition of right-up step-line and down-right step-line of c.

Similarly, the claim holds for the other cases. $\square$

3 From a Sequence-Pair to a Packing

In the previous section, we analyzed the packing, and fixed the procedure **Gridding** to get one sequence-pair from a given packing. Now we synthesize one packing from an arbitrary sequence-pair.

3.1 (Γ_+, Γ_-)-Packing

Let (Γ_+, Γ_-) be an arbitrary sequence-pair. We define a geometrical constraint derived from (Γ_+, Γ_-).

GeomConst(Γ_+, Γ_-)

> For every two modules x and x', x' must be right to x in Π if $x' \in \mathcal{M}^{\mathrm{aa}}(x)$. This is also the constraint with replacing the pair of words ("$\mathcal{M}^{\mathrm{aa}}$" and "right to") with any of ("$\mathcal{M}^{\mathrm{bb}}$" and "left to"), ("$\mathcal{M}^{\mathrm{ba}}$" and "above"), and ("$\mathcal{M}^{\mathrm{ab}}$" and "below").

A packing Π is called (Γ_+, Γ_-)*-packing* if Π satisfies **GeomConst**(Γ_+, Γ_-).

Corollary 1 : There is a (Γ_+, Γ_-)-packing if (Γ_+, Γ_-) is the sequence-pair obtained by **Gridding**. $\square$

However, we can prove the following fact.

Theorem 2 : For every sequence-pair (Γ_+, Γ_-), there is a (Γ_+, Γ_-)-packing.

Proof: Consider an $m \times m$ grid. Label the horizontal grid lines and vertical grid lines with module names along Γ_+ and Γ_- from top and from left in order, respectively. A cross point of the horizontal grid line of label x and the vertical grid line of label x' is referred to by (x, x'). Then, draw the resultant grid on a plane rotating 45 degrees. (See Fig.4.) Put each module x with its center being on (x, x). Expand the separation of grid lines enough to eliminate overlapping of modules. (Actually, the expansion is enough if it is $\sqrt{2}$ times larger than the longest width/height over modules.) Trivially, the resultant packing satisfies the requirement implied by the given sequence-pair. $\square$

An example is shown in Fig.4 which corresponds to $(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$.

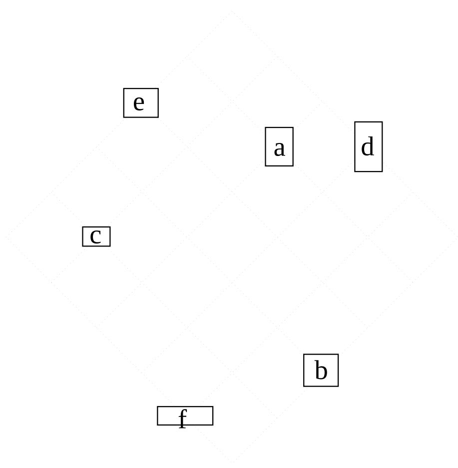


Fig.4: A (Γ_+, Γ_-)-packing
$(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$

3.2 (Γ_+, Γ_-)-Optimal Packing

Given (Γ_+, Γ_-), an optimal packing of (Γ_+, Γ_-)-packings is said (Γ_+, Γ_-)-optimal. A (Γ_+, Γ_-)-optimal packing can be obtained in $O(m^2)$ time by applying the well-known *longest path algorithm* for vertex weighted directed acyclic graphs. The formulation is given below.

Based on "right to" constraint of $(\Gamma+, \Gamma_-)$, a directed and vertex-weighted graph, called *horizontal-constraint graph* $G_H(V, E)$, is constructed as follows.

V : source s, sink t, and m vertices labeled with module names

E : (s, x) and (x, t) for each module x, and (x, x') iff x and x' are in the constraint "x must be left-to x'".

Vertex-weight : zero for s and t, width of module x for the other vertices

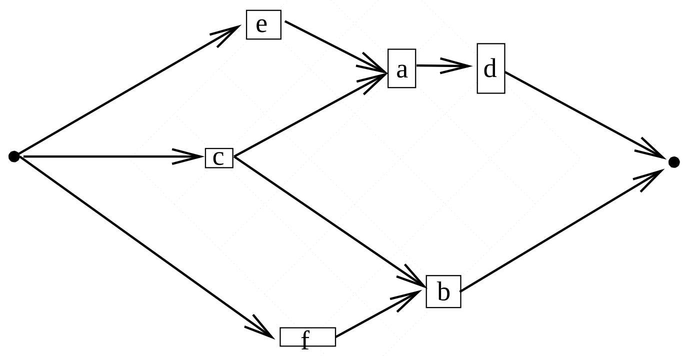
Similarly the *vertical-constraint graph* $G_V(V, E)$ is constructed using "below" constraint and the height of each module.

Both graphs do not contain any directed cycle. Furthermore, for every pair of modules, there is always an edge in G_H or in G_V, and not in both. From this fact, the X-coordinate and the Y-coordinate of each module can be determined independently to satisfy the constraint in the direction, and the resultant placement is guaranteed not to contain any overlap. Then, the X and Y coordinates of each module are determined as the minimum by assigning the longest path length between the source and the node of the module in G_H and G_V, respectively. Similarly, the width and the height of the chip is determined as the longest path length between the source and the sink in G_H and G_V, respectively. Since the width and the height of the chip is independently minimum, the resultant packing is (Γ_+, Γ_-)-optimal. The longest path calculation can be done in $O(m^2)$ time, proportional to the number of edges in the graph.

As an example, G_H and G_H are shown in Fig.5 for $(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$. The resultant placement after longest path length calculation is shown in Fig.6.

3.3 The P-admissible Solution Space

Previous discussions conclude:



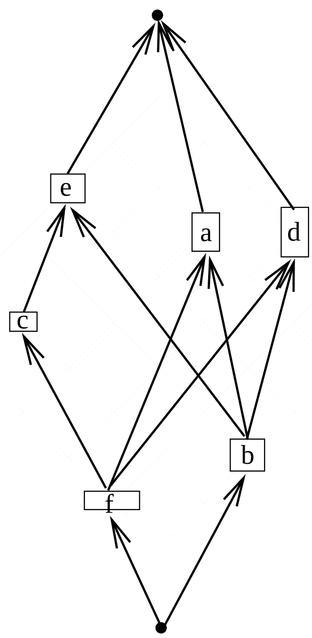


Fig.5: Constraint graphs G_H(above) and G_V(below) (transitive edges are not drawn for simplicity)

Theorem 3 : The set of all sequence-pairs is a P-admissible solution space of **RP**. More precisely, it consists of $(m!)^2$ sequence-pairs, each of which can be mapped to a packing in $O(m^2)$ time, and at least one of which corresponds to an optimal solution of **RP**. $\square$

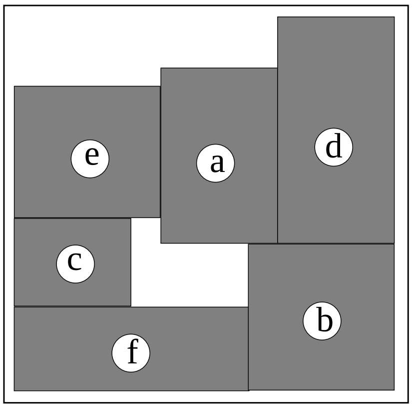


Fig.6: A (Γ_+, Γ_-)-optimal packing
for $(\Gamma_+, \Gamma_-) = (ecadfb, fcbead)$

Our discussion started for minimizing the area of the chip. However, all the theorems except Theorem 3 do not mention about the evaluation. While, Theorem 3 holds for any evaluating function as long as it is independently non-decreasing both for the width and height of the chip. Therefore we may assume instead, for example, perimeter of the chip, area of chip of pre-specified aspect ratio, and height of the chip when width of the chip is fixed. This fact will expand the applicability of our solution space.

It has also been assumed that the orientation of each module (vertically laid or horizontally laid) is fixed. When the orientation is also requested to be optimized, we hold a $\{0, 1\}$ sequence of length m, expressing the orientation of each module is horizontal or vertical. The solution space is enlarged to the size of $(m!)^2 2^m$. (The orientation optimization for a fixed floorplan is known to be NP-hard[11].) This technique can be easily extended for so called "soft" modules, by preparing three or more candidates of (width, height) pairs for one module[12].

4 Experiments

4.1 Rectangle Packing

To show the usefulness of the proposed solution space, dimensions of 146 modules are extracted from a printed circuit board in an industry example, and packed by a simulated annealing method based on three kinds of pair-interchanges of: (i) two module names in Γ_+, (ii) two module names both in Γ_+ and Γ_-, and (iii) the width and the height of a module, where the last one is for orientation optimization. The initial sequence-pair was given by assuming $\Gamma_+ = \Gamma_-$. The temperature was decreased following a quite standard annealing schedule but from a heuristic point of view, operation (i) was selected with higher probability in higher temperature, and operation (iii) was selected with higher probability in lower temperature.

The result is shown in Fig.7. Computation on Sun SparcStationII stopped in 29.9 minutes reaching the terminating temperature. The algorithm has searched not greater than 606,192 distinct sequence-pairs out of the solution space of size $(146!)^2 2^{146} \sim 1.23 \times 10^{552}$. Notice that, only the search of a fraction about 4.92×10^{-547} of the solution space was enough to obtain Fig.7. As a challenge, we tried 500 pieces, using 18.83 hours to get the result shown in Fig.8.

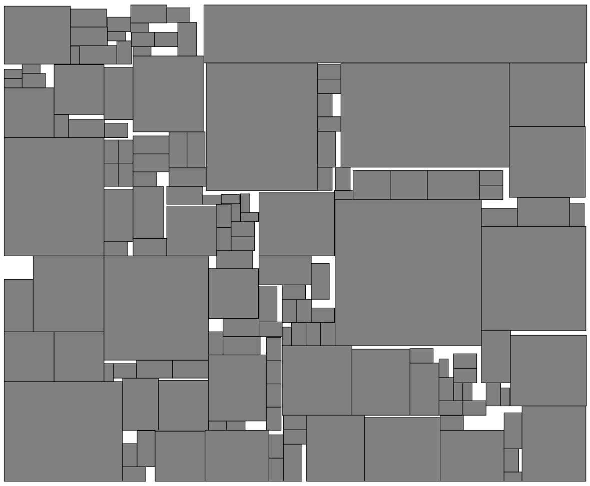


Fig.7: Packing of 146 modules

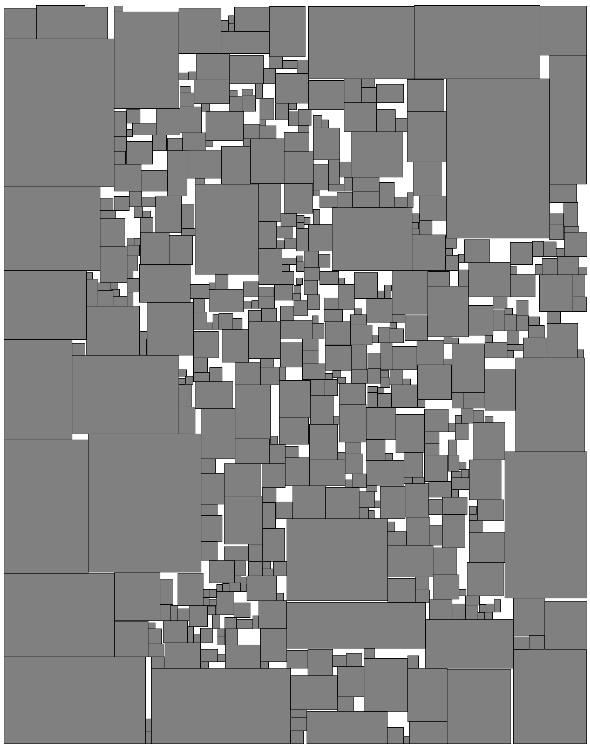


Fig.8: Packing of 500 modules

4.2 Module Placement with Wire Consideration

For VLSI placement problems, we extend the evaluation to consider wires. Among various possible evaluations about wires, here we focus the final chip area (after every wires are routed in detail). It would be ideal to invent a

P-admissible solution space for this evaluation. However, it is difficult at this moment and we make an estimation of final chip area without actual routing phase.

Assume (Γ_+, Γ_-) be a sequence-pair, Π be a (Γ_+, Γ_-)-optimal packing, and W and H be the width and height of Π. Terminals are given as fixed points on the boundary of each module. A *net* is a set of terminals(multi-terminal net), which must be connected by wires, later in detailed routing phase. A set of net is given as the netlist $\mathcal{N}$. When observing the terminals of a net i, which spread over Π, the width and height of the smallest bounding box of these terminals are denoted W_i and H_i, respectively. T is the sum of wiring width and wiring space (obtained from a design rule set). We use following formulae to estimate the final chip width W' and height H', which is the one proposed in [10].

$$\begin{aligned} W' &= W + T\frac{\Sigma_{i\in\mathcal{N}} H_i}{H} \\ H' &= H + T\frac{\Sigma_{i\in\mathcal{N}} W_i}{W} \end{aligned}$$

The second term of each formula estimates the amount of increase in one direction owing to the wires, assuming all wires are uniformly distributed in the final chip. They experimentally showed that the resultant placement is acceptable for later detailed routing stage.

There are totally eight choices per a module, which is the combination of four choices of $\{0, 90, 180, 270\}$ degree rotations, and two choises of {yes,no} value of the reflection about Y axis. In our system, this code for orientation and a sequence-pair are put together into a simulated annealing process, which runs in a similar fashon as rectangle packing optimization. The process searches the solution space of size $(m!)^2 8^m$.

The point which is not mentioned in [10] is how the location of each individual module is calculated. After the best evaluated code is obtained, coordinates of each module is determined as follows. Assume (X_j, Y_j) is the coordinates of lower left corner of module j in Π. (Which is the information we can use in this phase.) Let $\mathcal{N}_{X_j}$ be a set of nets such that the X coordinate of left side of bounding box of the net is less than or equal to X_j. Similarly, $\mathcal{N}_{Y_j}$ is defined using Y_j. We determine the coordinates (X'_j, Y'_j) of the lower left corner of module j in the resultant chip by the following formulae.

$$\begin{aligned} X'_j &= X_j + T\frac{\Sigma_{i\in\mathcal{N}_{X_j}} H_i}{H} \\ Y'_j &= Y_j + T\frac{\Sigma_{i\in\mathcal{N}_{Y_j}} W_j}{W} \end{aligned}$$

The biggest building block layout data, called "ami49", is taken from the MCNC benchmarks, and placed by the above described method in a one level computation with additional constraint: aspect ratio = 1. For authors' knowledge, it has not been dealt without hierarchically dividing the problem to reduce the problem size. The result is shown in Fig.9. Computation time was 31.36 minutes on SunIPX.

5 Concluding Remarks

The motivation of this work was our experience that many VLSI designers are not satisfied with slicing structures. This paper has achieved a breakthrough by introducing a P-admissible solution space to the rectangle packing problem, which is fundamental to the layout design.

Experiments suggest that hundreds of rectangles can be packed very effectively in reasonable time. The biggest MCNC benchmark data, ami49, is now tractable without hierarchically dividing the problem, by utilizing the proposed solution space.

As experiments revealed, the search covers only a fraction whole of the solution space. Though the results are of the quality high enough for practical use, the space is too vast. They would hope to create a smaller space but we conjecture that the size of the space can not be decreased drastically.

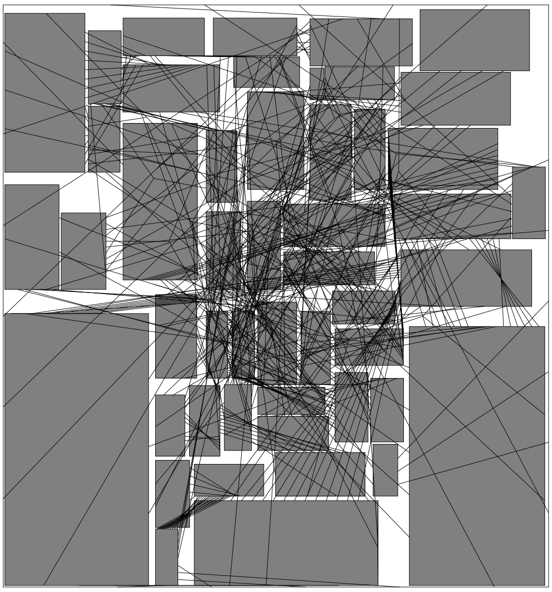


Fig.9: Placement of MCNC "ami49"

References

[1] B.S.Baker, E.G.Coffman, and R.L.Rivest, "Orthogonal Packings in Two Dimensions," *SIAM J. Comput.*, vol. 9, no. 4, pp. 846–855, 1980.

[2] L. Sha and R. W. Dutton, "An Analytical Algorithm for Placement of Arbitrarily Sized Rectangular Blocks," in *Proc. 22th ACM/IEEE Design Automation Conf.*, pp. 602–608, 1985.

[3] A.Alon and U.Ascher, "Model and Solution Strategy for Placement of Rectangular Blocks in the Euclidean Plane," *IEEE Trans. on CAD*, vol. 7, no. 3, pp. 378–386, 1988.

[4] Y.G.Saab and V.B.Rao, "Combinatorial Optimization by Stochastic Evolution," *IEEE Trans. on CAD*, vol. CAD-10, no. 4, pp. 525–535, 1991.

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

[6] 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.

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

[8] W.M.Dai and E.Kuh, "Simaltaneous Floorplanning and Global Routing for Hierarchical Building Block Layout," *IEEE Trans. on CAD*, vol. CAD-6, no. 5, pp. 828–837, 1987.

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

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

[11] L.Stockmeyer, "Optimal Orientations of Cells in Slicing Floorplan Designs," *Information and Control*, vol. 59, pp. 91–101, 1983.

[12] P. Pan, W. Shi, and C. L. Liu, "Area Minimization for Hierarchical Floorplans," in *IEEE International Conf. on Computer Aided Design*, pp. 436–440, 1994.