社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2004－SLDM－117（17）
2004／12／2

3D BSG 構造を用いた 3 次元パッキング表現手法

山岸　弘和†　二宮　洋††　浅井　秀樹†

† 静岡大学システム工学科　〒432-8561 静岡県浜松市城北 3-5-1
†† 湘南工科大学情報工学科　〒251-8511 神奈川県藤沢市辻堂西海岸 1-1-25

あらまし　近年 VLSI の微細化によって 3 次元 Chip デザインが重要視されるようになってきた。このため、複数の直方体を 3 次元空間上に最小の体積となるように重ならずに配置する 3 次元パッキング問題が重要視されてきている。我々はこれまでに、3 次元パッキングを行うための表現手法として 3DBSG (three-dimensional Bounded-Sliceplane Grid) を提案してきた。3DBSG によって各直方体に 3 次元空間内のそれぞれの 3 つの位置関係 "right-of", "rear-of", "above" を与えられることにより重ならない配置を計算することが出来る。本稿では、L 型の 3 次元モジュールを含む 3 次元パッキングを 3DBSG 構造を用いて表現することを考える。また、これを検証するために標準的な Simulated Annealing を最適化法として採用したシミュレーションを行い、提案手法の有効性を検証する。
キーワード　3 次元パッキング, 3DBSG, L 型モジュール, シミュレーティッド・アニーリング

Three Dimensional Module Packing using 3DBSG structure

Hirokazu YAMAGISHI†, Hiroshi NINOMIYA††, and Hideki ASAI†

† Department of Systems Engineering, Shizuoka University 3-5-1, Johoku, Hamamatsu 432-8561 Japan
†† Department of Information Science, Shonan Institute of Technology 1-1-25, Nishikaigan, Tsujido, Fujisawa, 251-8511, Japan

Abstract　A three-dimensional (3D) chip design strategy will be of increasing significance in association with miniaturization of VLSI circuits. In the design of 3D VLSI, the placement which is to locate given 3D rectangular objects without overlapping each other in the 3D space, becomes an important issue. The 3D placement under the nonoverlapping constraint is called the 3D packing in which minimizing the volume of the package is required. This paper introduces a new coding system to encode the topology of 3D packing as an extension of BSG(Bounded-Sliceline Grid) for 2D packing. The novel coding system is referred to as three dimensional Bounded-Sliceplane Grid(3DBSG). The topology is a set of relative relations assigned to pairs of 3D modules such that "right-of", "rear-of" and "above". We further show idea to handle L-shaped modules on the 3DBSG. The simulation results in which a standard simulated simulated annealing is utilized as an optimization algorithm, are demonstrated in order to test the validity of our 3D packing method based on 3DBSG.
Key words　3D Packing, 3DBSG, L-shaped modules, simulated annealing

1. Introduction

Performance of deep-submicrometer very large scale integrated(VLSI) circuits is being increasingly dominated by interconnects due to decreasing wire pitch and increasing die size. Additionally, heterogeneous integration of different technologies in one single chip is becoming increasingly desirable, for which two-dimensional (2D) ICs may not be suitable [1]. A three-dimensional (3D) chip design strategy will be of increasing significance in association with the miniaturization of VLSI circuits in the immediate future. In the design of 3D VLSI, the placement which is to locate given 3D rectangular objects without overlapping each other in the 3D space, becomes an important issue. The 3D placement under the nonoverlapping constraint is called the 3D packing in which minimizing the volume of the packing is required.

Additionally, 3D placement has been studied in the research of Field Programable Gate Arrays(FPGAs) which have the partially dynamic reconfigurable architectures [2]

A small number of previous works have been reported on

3D packing with heuristic optimization approaches [2]- [7]. In [3], 3D packing problem aimed at fixed capacity box like a container or a warehouse has been investigated. A neuro-based optimization algorithm has been utilized to solve 3D puzzles in [4] [5]. In [2] [7], 3D packing algorithms based on the tree representations have been introduced. The paper [6] has extended the coding system for 2D packing which is called the sequence-pair [8] to handle 3D packing problem. The coding system for 3D packing has been referred to as the sequence-triple. The sequence-triple can be decoded to the unique topology and packing of modules. Here the topology is a set of relative relations assigned to pairs of modules such that "right-of", "rear-of" and "above". It has been easily to implement a stochastic search algorithm for the 3D packing problem based on this coding system. A hundred of 3D rectangular solid modules have been satisfiably packed within several hours.

On the other hand, Nakatake et al. in [9] presented a grid based representation-BSG(Bounded Sliceline Grid) for 2D packing. The BSG structure utilizes a set of horizontal and vertical bounded-length lines called Segs to cut the plane into rooms and represents a placement using two directed graphs, horizontal and vertical graphs, created by these lines and rooms. 2D packing algorithm based on BSG structure has been very flexible in practical purpose.

This paper describes a novel strategy for 3D packing with an extended BSG structure which is referred to as 3DBSG(Three Dimensional Bounded Sliceplane Grid) [10]. In the 3DBSG, 3D space is dissected by the squares which are a set of the x, y and z directions bounded-planes, called 3DBSG-Squs, and split up into cubes, called 3DBSG-Rooms. In a manner similar to BSG, the topology of the modules is represented by using three directed graphs which are applied along three orthogonal axes, the x-, y- and z-axis. Here, the x-, y- and z- directed graphs indicate the relative relations of two modules such that "right-of", "rear-of" and "above", respectively. Here, we introduce an extra 3DBSG-Squ because the above three 3DBSG-Squs can not satisfy the nonoverlapping constraint. Furthermore an idea to handle the L-shaped 3D modules in 3DBSG structure is discussed. Finally, our proposed method is compared with one based on sequence-triple for 3D packing. It is shown that the method proposed here can calculate the 3D placement faster than the method based on sequence-triple without deteriorating the quality of a solution through the computer simulations. A standard simulated annealing algorithm is utilized as an optimization method. As a result, it is confirmed that the 3D placement based on 3DBSG structure is efficient and practical for the 3D module packing.

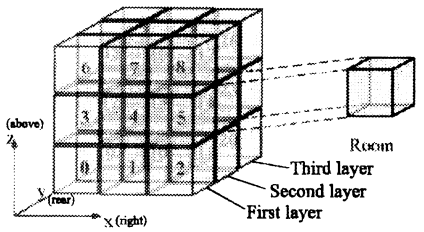


Fig. 1 3D space with (x,y,z)-coordinate system and $3DBSG_{3\times3\times3}$.

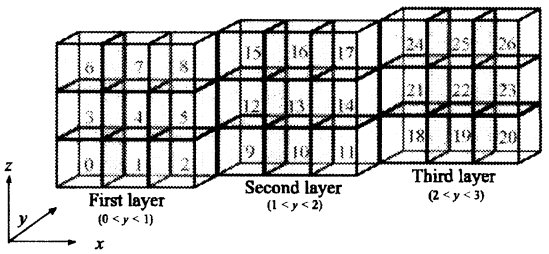


Fig. 2 3DBSG-Rooms.

2. 3-Dimensional(3D) Packing based on 3DBSG

In this section, a novel strategy for 3D Packing with an extended BSG structure is proposed. This structure is referred to as 3 Dimensional Bounded Sliceplane Grid(3DBSG).

2.1 3DBSG structure

In the 3D packing, (x, y, z)-coordinate is correlated to a set of relative relations assigned to pairs of boxes such that the x-, y- and z-axis indicate "right-of", "rear-of" and "above" relations, respectively as shown in Fig.1. 3D space in (x, y, z)-coordinate system is dissected by the squares which are a set of the x, y and z directions bounded-planes 2 on a side called 3DBSG-Squs(or simply Squs), and split up into cubes 1 on a side, called 3DBSG-Rooms(or simply rooms). Example of 3DBSG-Table with 3 × 3 × 3 3DBSG-rooms represented by $3DBSG_{3\times3\times3}$, is shown in Fig.1 and Fig.2. In Fig.2, 3DBSG-Rooms are expanded as the layers to the y-directions, that is, from front to rear, and each room is labeled to a linear order. Three Squs are defined as

$$W_{i,j,k} = \left\{ (x,y,z) \;\middle|\; \begin{array}{c} x = i, j-1 < y < j+1, \\ k-1 < z < k+1 \\ \text{if } i : even \text{ then } j,k : odd \\ \text{if } i : odd \text{ then } j,k : even \end{array} \right\}, \quad (1)$$

$$D_{i,j,k} = \left\{ (x,y,z) \;\middle|\; \begin{array}{c} y = j, k-1 < z < k+1, \\ i-1 < x < i+1 \\ \text{if } j : even \text{ then } i,k : even \\ \text{if } j : odd \text{ then } i,k : odd \end{array} \right\}, \quad (2)$$

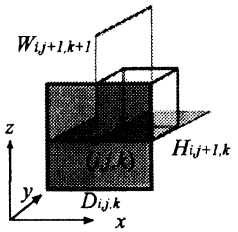


Fig. 3 3DBSG-Room and Squs.

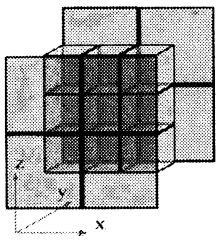


Fig. 4 Example of the depth Squs for the first layer.

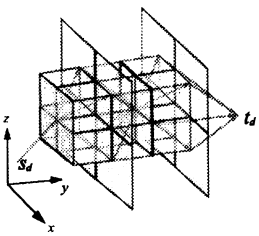


Fig. 7 The depth graph on 3DBSG.

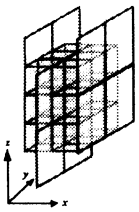


Fig. 5 Example of the width Squs.

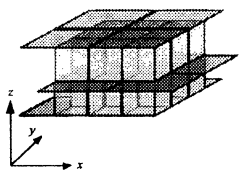


Fig. 6 Example of the height Squs.

$$H_{i,j,k} = \left\{ (x,y,z) \;\middle|\; \begin{array}{c} z = k, i-1 < x < i+1, \\ j-1 < y < j+1 \\ \text{if } k\text{:}even \text{ then } i\text{:}even, k\text{:}odd \\ \text{if } k\text{: } odd \text{ then } i\text{: } odd, k\text{: } even \end{array} \right\} \quad (3)$$

where $W_{i,j,k}$, $D_{i,j,k}$ and $H_{i,j,k}$ are called 'width Squ', 'depth Squ' and 'height Squ', respectively, and the subscripts (i,j,k) denote the (x,y,z)-coordinate of the center of Squs. The bottom-left corner of the front plane of the room labeled '0' is the origin of (x,y,z)-coordinate system. Two depth Squs $D_{i,j,k}$ and $D_{i',j',k'}$ are said to be "adjacent" if $|i-i'|=1$, $|j-j'|=1$ and $|k-k'|=1$. Furthermore the relation between $D_{i,j,k}$ and $D_{i',j',k'}$ is that $D_{i',j',k'}$ is "rear-of" $D_{i,j,k}$, if $j'=j+1$, $|i-i'|=1$ and $|k-k'|=1$. For the width Squs, similar relations "adjacent" and "right-of" are defined. The "adjacent" and "above" relations are also given by the height Squs. Three Squs, $W_{i,j+1,k+1}$, $D_{i,j,k}$ and $H_{i,j+1,k}$, which are assigned to a room are illustrated in Fig.3. Here, the coordinate of the room, (i,j,k), is set to the bottom-left corner of the front plane. An example of "depth-Squs" for the first layer is illustrated in Fig.4. As shown in Fig.4, the depth-Squs are put parallel in (x,z)-coordinate systems. Similarly, the width Squs and the height Squs are oriented parallel with (y,z)- and (x,y)-coordinate systems, respectively. The width Squs and the height Squs are shown in Fig.5 and Fig.6, respectively.

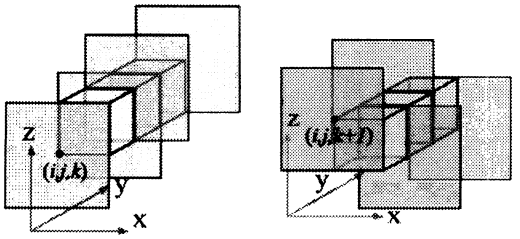
2.2 Directed graphs

Similarly to the BSG, three directed graphs, that is, width, depth and height graphs are defined as $G_w(V_w,E_w)$, $G_d(V_d,E_d)$ and $G_h(V_h,E_h)$, respectively, in the $3DBSG_{p\times q\times r}$.

Let $V_d=\{s_d,t_d\}\cup\{u_{i,j,k}\}$ where $u_{i,j,k}$ corresponds to the center of Squ $D_{i,j,k}$. Edges are defined as follows. s_d is the source connected to all the vertices corresponding to the front Squs, i.e., $D_{0,0,0}$, $D_{2,0,0}$, $D_{0,0,2}$, $D_{2,0,2}$, ..., $D_{2i,0,2k}$ where $i=\lfloor(p-1)/2\rfloor$ and $k=\lfloor(r-1)/2\rfloor$. t_d is the sink connected from all the vertices corresponding to the rear Squs which are, for example, for the case q : $even$, $D_{0,q,0}$, $D_{2,q,0}$, $D_{0,q,2}$, ..., $D_{2i,q,2k}$ where $i=\lfloor(p-1)/2\rfloor$ and $k=\lfloor(r-1)/2\rfloor$. The other edge $(u_{i,j,k},\ u_{i',j',k'})$ exists if $D_{i',j',k'}$ is "rear-of" and "adjacent" to $D_{i,j,k}$. The directed depth graphs are shown in Fig.7. The width and height directed graphs, $G_w(V_w,E_w)$ and $G_h(V_h,E_h)$ are similarly defined. Here the relative relation assigned to pairs of modules such that "rear-of" are defined as $G_d(V_d,E_d)$. Similarly, the "right-of" and "above" relations are derived from $G_w(V_w,E_w)$ and $G_h(V_h,E_h)$, respectively.

2.3 Extra depth Squs and directed graphs

There are some rooms which can not be associated with other rooms by the above three Squs and directed graphs. An example is described as follows. Here the room labeled '13' which is the core room of $3DBSG_{3\times3\times3}$ in Fig.1 is considered. The room '13' is surrounded by 6 Squs that are $D_{1,1,1}$, $D_{2,2,2}$, $W_{1,2,2}$, $W_{2,1,1}$, $H_{1,2,1}$ and $H_{2,1,2}$. The "rear-of" relation between the room '13' and each room labeled '0', '1', '3' and '4' is given by the depth Squ $D_{1,1,1}$. Furthermore the rooms labeled '22', '23', '25' and '26' are "rear-of" the room '13', where this relation is derived from the depth Squ $D_{2,2,2}$. Similarly, the width Squs, $W_{1,2,2}$ and $W_{2,1,1}$ are used to connect the room '13' and rooms '12', '15', '21', '24', '2', '5' '11' and '14'. The height Squs, $H_{1,2,1}$ and $H_{2,1,2}$ relate the room '13' to rooms '9', '10', '18', '19', '7', '8', '16' and '17'. As mentioned above, the room '13' is associated with the 24 rooms by using width, depth and height Squs except for the rooms labeled '6' and '20'. This condition violates the nonoverlapping constraint. Therefore we introduce an extra depth Squ, $D'_{i,j,k}$ as follows:



(a) The depth Squs. (b)The extra depth Squs.

Fig. 8 The depth Squs and the extra depth Squs.

$$D'_{i,j,k} = \left\{ (x,y,z) \;\middle|\; \begin{array}{c} y = j, k-1 < z < k+1, \\ i-1 < x < i+1 \\ \text{if } j\text{:}even \text{ then } i\text{:}even, k\text{:}odd \\ \text{if } j\text{:}odd \text{ then } i\text{:}odd, k\text{:}even \end{array} \right\} \quad (4)$$

Moreover an extra directed depth graph is also defined as $G'_d(V'_d, E'_d)$. The extra depth Squs are illustrated in Fig.8. As shown in Fig.8(b), an extra depth Squ is put in $(i, j, k+1)$, if the coordinate of the depth Squ is (i, j, k). By using $D'_{1,1,2}$ and $D'_{2,1,2}$, the rooms '13' can be related to '3', '4', '6', '7', '19', '20', '22' and '23'. As a result, the room '13' can be associated with the other 26 rooms by using four types Squs. In this paper only the extra Squs for depth direction is introduced to keep the nonoverlapping constraint. The proof for the existence of an optimal solution is outside the scope of this paper.

2.4 Placement

In this subsection, 3D packing strategy based on the 3DBSG is introduced by using the similar method of the BSG. On a 3DBSG-assignment, each module is assigned into a 3DBSG-Room without overlapping. An example of 3DBSG-assignment is shown in Fig.9. A room to which no module is assigned, is said "empty room". The edges in $G_w(V_w, E_w)$ and in $G_h(V_h, E_h)$ are weighted by the width and the height of assigned modules, respectively. Both edges in $G_d(V_d, E_d)$ and $G'_d(V'_d, E'_d)$ are weighted by the depth of assigned modules. The weights of the edges crossing the empty room and connecting the source or sink are set to 0.

Let $l_w(u)$ be the length of the longest path from source s_w to $u \in V_w$ in $G_w(V_w, E_w)$. In the same way, $l_h(u)$ denotes the longest path from source s_h to $u \in V_h$ in $G_h(V_h, E_h)$. Furthermore $l_d(u)$ is defined as the longer path either the longest path from source s_d to $u \in V_d$ in $G_d(V_d, E_d)$ or the longest path from source s_d to $u \in V'_d$ in $G'_d(V'_d, E'_d)$, where the paths from both of V_d and V'_d cross the same room. The (x, y, z)-coordinate of module m which is assigned into a room whose boundary left width Squ is W, bottom height Squ is H and front depth or extra depth Squ is D, is given by using $l_w(u_W)$, $l_h(u_H)$ and $l_d(u_D)$. Furthermore the size of 3D packing area, V_{size} is evaluated by

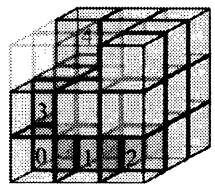


Fig. 9 Example of the 3DBSG-assignment.

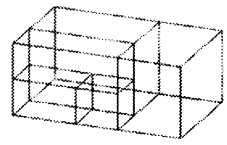


Fig. 10 The result of the optimal placement.

$$V_{size} = l_w(t_w) \times l_d(t_d) \times l_h(t_h). \quad (5)$$

The optimal packing result for an example of 5 modules whose (width, depth, height)-data are m0(3,2,2), m1(2,2,2), m2(3,4,5), m3(5,2,2) and m4(5,4,3), is illustrated in Fig.10.

2.5 Optimization algorithm for 3D packing

We can gather from the fact that 2D packing problem is NP-hard [8] that 3D packing problem is also NP-hard. Therefore a standard simulated annealing algorithm is utilized as an optimization algorithm. In the simulated annealing, a new solution is derived from an interchange of the contents of two rooms, at least one of which is nonempty. Moreover, 90 degee rotation of the rectangular solid is considered in this algorithm.

2.6 Packing of L-shaped 3D Modules

In [9], it has been introduced that the BSG structure can handle the rectilinear modules, typically, L-shaped modules for the 2D packing, without significant extra computation. Here we propose an idea to handle the L-shaped 3D modules on the 3DBSG. An example of the L-shaped 3D modules is shown in Fig.11(a). Procedure of embedding an L-shaped 3D module is illustrated in Fig.11. Cut the module into two pieces, that is, S_1 and S_2 by the middle square abut on the bottom and front squares shown in Fig.11(b). By regarding the middle, the front and the bottom squares as the x-, y- and z-Squs, respectively, the S_1 and S_2 modules can be assigned to a pair of rooms an shown in Fig.11(c). In the simulated annealing, an additional condition, that is, these two modules are always moved together, is imposed.

3. Experimental results

We perform two sets of experiments: one is rectangular modules packing, and the other is 3D packing including on some artificial L-shaped 3D modules.

3.1 Rectangular solid module packing

In 2D packing based on the BSG, an n by n grid plane has been theoretically needed for the optimal placement of n modules, whereas the empirical compromise has been discussed in [9] to obtain a suitable solution for the practical use. The same holds for 3D packing based on the 3DBSG.

First, we consider reducing the number of rooms to ob-

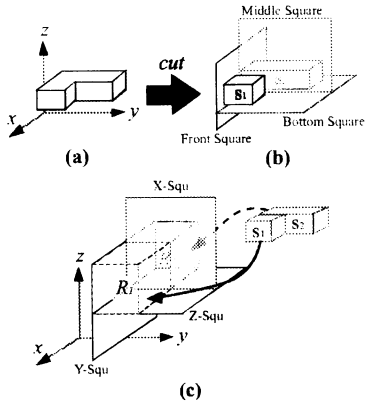


Fig. 11 Procedure of embedding an L-shaped 3D module.

tain the packing solutions in the practical time without losing the quality of solutions. To explore the relations among "quality" of the solution, "3DBSG-table size" and "computation time", the following simulations are performed. We prepare 27 modules of random size within [1, 128] and various tables $3DBSG_{3\times3\times3}$, $3DBSG_{4\times4\times4}$, ..., $3DBSG_{16\times16\times16}$ and $3DBSG_{27\times27\times27}$. The simulation results are shown in Fig.12. The standard simulated annealing in which 500,000 iterations are performed, is used as an optimization algorithm for 3D packing. Moreover the volume of packing and the computation time using $3DBSG_{27\times27\times27}$ are 7902924 and 607343[ms], respectively. From Fig.12(a), it is shown that there is almost no variation with respect to "quality" of the solutions even though various size of 3DBSG-table are used, except for $3DBSG_{3\times3\times3}$. From Fig.12(b), the computation time grows exponentially with an increase in 3DBSG-table size. As a result, it is observed that if the number of rooms, r is within $(\sqrt[3]{n}+1)^3 < r < (3\sqrt[3]{n})^3$, the quality of the solution is held. The above observation can be confirmed through the other examples.

Second, the computer simulations are conducted in order to test the validity of the proposed 3D packing strategy, where the standard simulated annealing with 500,000 iterations is also utilized as an optimization algorithm. In this simulation, 100 modules whose sizes are generated by random integer in [1,128], are treated as an example. The theoretical minimum value, V_{th} is 25885848 which is sum of all module volumes. 100 simulation runs are performed with the different initial solutions. The simulation results are illustrated in TABLE 1, where avg, max and min denote the volume of packing of average, of maximum and of minimum, respectively, and the number of rooms, r is $(3\sqrt[3]{n})^3$. The simulations are performed on the computer with Athlon XP 2100+ CPU, 512 MB memory and Windows 2000. From TABLE 1, it is shown that the proposed method can obtain

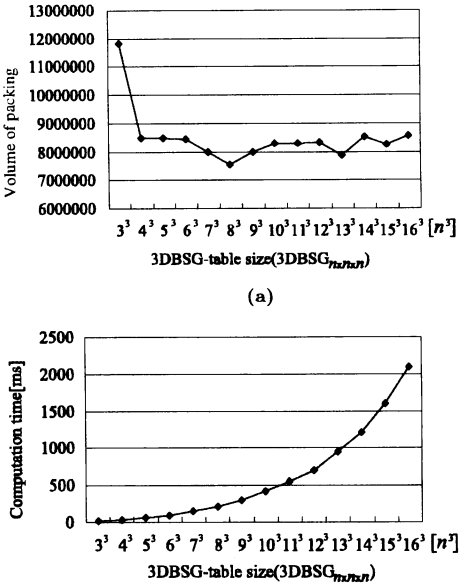


Fig. 12 Relations among "quality" of the solution, "3DBSG-table size" and "computation time".

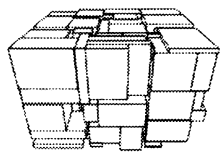


Fig. 13 The result of the 100 modules packing.

the solution with volume ratio(volume of packing / sum of all module volumes, V_{th}) of 1.26. The result of 3D packing with 100 modules is shown in Fig.13.

Finally, we compare the proposed 3D packing method with one based on the sequence-triple [6] using three examples which are 27, 49 and 100 modules. The iteration counts of the simulated annealing which are same for all simulations, are 500,000. The simulations are performed on the computer with Pentium4 2.20GHz CPU, 2GB memory and Windows XP. The simulation results are shown in TABLE 2. From TABLE 2, it is confirmed that 3D packing based on the 3DBSG is superior to one based on the sequence-triple from the viewpoint of the simulation speed without losing the quality of the solutions, if the suitable 3DBSG-table size is selected. As a result, we found that our method is efficient and practical for the 3D packing.

3.2 3D module packing including L-shaped 3D modules

Here we test 10 rectangular solids and 5 L-shaped 3D modules packing problem. The theoretical minimum volume, V_{lth}

Table. 1 Simulation results for 100 modules.

	volume of packing	volume of packing / V_{lth}	time[ms]
avg	35126387	1.3569726	481740
max	37354070	1.4430306	495625
min	32689300	1.2628252	467562

Table. 2 Comparison of 3DBSG with sequence-triple.

boxes	sequence-triple		3DBSG		
	volume	time[ms]	# of rooms, r	volume	time[ms]
27	7440000	130203	5^3	7469280	57734
			9^3	7675335	300421
49	15732873	356046	5^3	15734499	58375
			9^3	14596200	302296
100	36128435	1509843	6^3	34094248	99531
			12^3	34388640	713219

Table. 3 L-Shape 3D module packing

	$3DBSG_{7\times7\times7}$		$3DBSG_{11\times11\times11}$	
	V_{size}/V_{lth}	time[ms]	V_{size}/V_{lth}	time[ms]
avg	1.298266	29.2130	1.297014	108.7978
max	1.440974	30.1720	1.445661	113.2350
min	1.223972	29.0320	1.192976	108.0160

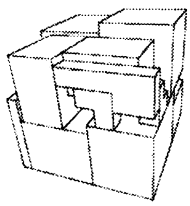


Fig. 14 The result of the 3D modules Packing include L-shaped 3D modules.

is 488181 which is sum of all module volumes. 100 simulation runs are performed with the different initial solutions on $3DBSG_{7\times7\times7}$ and $3DBSG_{11\times11\times11}$. The simulations are performed on the computer with Pentium4 2.20GHz CPU, 2GB memory and Windows XP. The simulation results are shown in TABLE 3. From TABLE 3, it is shown that the proposed method can obtain the acceptable solution with volume ratio(volume of packing / sum of module valumes, V_{lth}) of 1.19. The result of 3D modules packing is illustrated in Fig.14.

4. Conclusion

This paper has described a novel strategy for 3D module packing with an extended the BSG structure. The proposed structure has been referred to as the 3DBSG structure. Furthermore, we have proposed an idea to handle L-shaped 3D modules on the 3DBSG. The computer simulations have been conducted in order to test the validity of the proposed 3D packing strategy. The standard simulated annealing has been utilized as an optimization algorithm. As a result, we have found that our method has been efficient and practical for the 3D module packing.

References

[1] K.Banerjee, S.J.Souri, P.Kapur, and K.C.Saraswat "3-D ICs: A novel Chip Design for Improving Deep-Submicrometer Interconnect Performance and Systems-on-Chip Integration", Proc. of the IEEE, vol.89, no5, may 2001.

[2] P.H.Yuh, C.L.Yang, Y.W.Chan and H.L.Chen "Temporal Floorplanning Using 3D-subTCG", Proc. ASP-DAC'04, pp.725-730, Jan., 2004.

[3] T.Kawakami, M.Minagawa, and Y.Kakazu "Automatic Tuning of 3-D Packing Strategy and Rule-Base Construction Using GA", Trans. of IPSJ, vol.30, no6, pp761-768, 1992.

[4] H.Yamamoto, Y.Nakayama, H.Ninomiya, and H.Asai, "A Neuro-Based Optimization Algorithm for Three Dimensional Cyrindric Puzzles", IEICE Trans. Fundamentals, vol.E80-A, no6, pp.1049-1054, 1997.

[5] H.Yamamoto, H.Ninomiya, and H.Asai, "A Neuro-Based Optimization Algorithm for Rectangular Puzzles", IEICE Trans. Fundamentals, vol.E81-A, No.6, pp.1113-11118, 1998.

[6] H.Yamazaki, K.Sakanushi, S.Nakatake, and Y.Kajitani, "The 3D-Packing by Meta Data Structure and Packing Heuristics", IEICE Trans. Fundamentals, vol.E83-A, no4, pp.639-645, April 2000.

[7] H.Kawai and K.Fujiyoshi "3D-Block Packing using a Tree Representation", IEICE Technical Report, VLD2004-29, pp.49-54, June, 2004(in Japanese).

[8] H.Murata, K.Fujiyoshi, S.Nakatake, and Y.Kajitani, "VLSI module placement based on rectangle-packing by the sequence-pair", IEEE Trans. on CAD, vol.15, no12, pp.1518-1524, Dec. 1996.

[9] S.Nakatake, K.Fujiyoshi, H.Murata, and Y.Kajitani, "Module Packing Based on the BSG-Structure and IC Layout Applications", IEEE Trans. on CAD, vol.17, no6, pp.519-530, June 1998.

[10] H.Yamagishi, H.Ninomiya and H.Asai "The 3D Packing by using 3DBSG", IEICE Technical Report, NLP2004-10, pp.53-58, May, 2004(in Japanese).