Revisiting Floorplan Representations

Bo Yao, Hongyu Chen, Chung-Kuan Cheng, and Ronald Graham

Department of Computer Science and Engineering, University of California, San Diego

{byao, hchen, kuan, rgraham}@cs.ucsd.edu

ABSTRACT

Floorplan representations are a fundamental issue in designing floorplan algorithms. In this paper, we first derive the exact number of configurations of mosaic floorplans and slicing floorplans. We then present two non-redundant representations: a twin binary tree structure for mosaic floorplans and a slicing ordered tree for slicing floorplans. Finally, the relations between the state-of-the-art floorplan representations are discussed and their efficiency is explored.

1. INTRODUCTION

Floorplanning and building block placement are becoming more important for VLSI physical design, because circuit sizes are growing rapidly and hierarchical design with IP blocks is now widely used to reduce the design complexity.

Unfortunately, many floorplanning problems are NP-complete. Hence, most floorplanning algorithms adopt either analytical force directed methods or perturbations with random searches and heuristics. Because the efficiency and effectiveness of these opertaions rely on the expression of the geometrical relation between circuit blocks, floorplanning representation becomes a fundamental issue. The redundancy of the representations and the complexity of the transformation between a representation and its corresponding floorplan can determine the execution time and the quality of the results.

1.1 State-of-the-art in floorplan representations

The representation of floorplans has been intensively studied over the past few decades. Most of these representations have addressed the completeness of a floorplan topology. For a floorplan with a slicing structure [10], a binary tree representation was widely used. The leaves of the binary tree correspond to the blocks and each internal node defines a vertical or horizontal merge operation of the two descendents. The upper bound on the number of possible configurations for the tree is $O(n!2^{5n-3}/n^{1.5})$.

For a more general non-slicing floorplan, there was no efficient representation other than the constraint graph until the sequence pair[8] and the bounded-sliceline grid [9] were proposed in the mid 90's.

ISPD'01, April 1-4, 2001, Sonoma, California, USA.

In [8], Murata et al. proposed a sequence pair (SP) representation. They used two sets of permutations to represent the topological relations between blocks. Thus, the number of combinations of the SP's is $O((n!)^2)$. It takes O(nloglogn) time to transform between an SP and its corresponding placement [11].

In [9], Nakateke et al. introduced a bounded sliceline grid (BSG) approach. A special n-by-n grid is devised for placing n blocks. This approach has $n!C(n^2, n)$ combinations and contains a lot of redundancy. The time complexity of the transformation is $O(n^2)$.

More recently, Hong et al. [7] proposed a Corner Block list (CB) representation of a mosaic floorplan. The mosaic floorplan covers all slicing floorplans and all non-slicing floorplans with n rectangles for n blocks. This representation has a relatively small combination number $O(n!2^{3n-3}/n^{1.5})$ compared to SP or BSG, and the time complexity of the transformation is O(n).

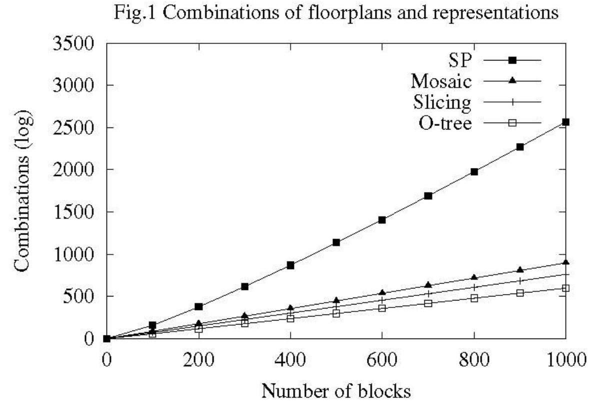
In [6], Guo et al. proposed an ordered tree (O-tree) representation of a non-slicing floorplan. An O-tree represents partial topological information, which together with the dimensions of all the blocks describes an exact floorplan. The O-tree has a very small combination number $O(n!2^{2n-2}/n^{1.5})$. The approach produces a floorplan in time O(n).

1.2 Our contributions

We find the exact numbers of slicing and mosaic floorplans. Two efficient representations of both slicing and non-slicing floorplans are developed. We then study the relationships between some of these representations. Our contributions in this paper include the following results:

(1) We find the exact numbers of two kinds of floorplans: mosaic floorplans and slicing floorplans. The number of mosaic floorplans is a Baxter number [1], and that of slicing floorplans is a Schröder number [5]. The combination numbers of SP's, mosaic floorplans,

Fig.1 Combinations of floorplans and representations



slicing floorplans, and O-trees are illustrated in a log scale below [Fig. 1]. The values are normalized by n!.

(2) We present two new representations of floorplans: a twin binary tree structure for mosaic floorplans and a slicing ordered tree structure for slicing floorplans. These two structures have zero redundancy because they present an exact one-to-one mapping to the corresponding floorplan.

(3) We study the relations between SP's, O-Trees, CBL's, twin binary trees, and slicing ordered trees. This study points out the redundancy of different representations.

Table 1: Exact Number of combinations of different floorplan configurations and representations

Nuber of blocks	Combinations of O-Tree	Combinations of Slicing floorplan	Combinations of Mosaic floorplan	Combinations of Sequence Pairs
1	1	1	1	1
2	2	2	2	2
3	5	6	6	6
4	14	22	22	24
5	42	90	92	120
6	132	394	422	720
7	429	1,806	2,074	5,040
8	1,430	8,558	10,754	40,320
9	4,862	41,586	58,202	362,880
10	16,796	206,098	326,240	3,628,800
11	58,786	1,037,718	1,882,690	39,916,800
12	208,012	5,293,446	11,140,560	479,001,600
13	742,900	27,297,738	67,329,992	6,227,020,800
14	2,674,440	142,078,746	414,499,438	87,178,291,200
15	9,694,845	745,387,038	2,593,341,586	1,307,674,368,000
16	35,357,670	3,937,603,038	16,458,756,586	20,922,789,888,000
17	129,644,790	20,927,156,706	105,791,986,682	355,687,428,096,000

2. NUMBERS OF FLOORPLANS AND REPRESENTATIONS WITH A ONE-TO-ONE MAPPING

In this section, we derive the exact numbers of floorplan configurations and describe the representations with a one-to-one mapping between the floorplan and its representation.

2.1 Exact number of mosaic floorplan configurations

Mosaic floorplanning was first introduced in [7], and has the following characteristics:

(1) There is no empty space within the floorplan, i.e., each rectangle is assigned to one and only one block. In the floorplan space, except the four corners of the chip, the segment intersection forms a T-junction. A T-junction is composed of a non-crossing segment and a crossing segment. The non-crossing segment has one end touching the crossing segment.

(2) The topology is equivalent before and after the non-crossing segment of the T-junction slides to adjust the block sizes.

(3) There is no degenerate case where two distinct T-junctions meet at the same point.

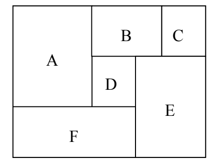


Fig. 2 An example of a mosaic floorplan

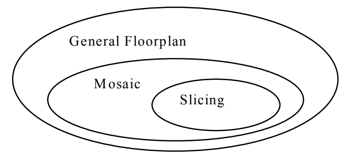


Fig 3. Categories of floorplans

Fig. 2 illustrates a mosaic floorplan example. In general, we can summarize that the set of slicing floorplans as a subset of the set of mosaic floorplans, which are in turn a subset of the set of general floorplans [Fig. 3].

The exact number of mosaic floorplan configurations turns out to be a Baxter number, which can be represented as follows:

$$B(n) = \binom{n+1}{1}^{-1} \binom{n+1}{2}^{-1} \sum_{k=1}^{n} \binom{n+1}{k-1} \binom{n+1}{k} \binom{n+1}{k+1} \quad (3.1)$$

These numbers were first used to count the number of Baxter permutations, which were first introduced in [1] in an attempt to prove a special function conjecture of Dyer. The following is their definition:

Definition 2.1.1 Baxter permutation: A Baxter permutation is a permutation $\sigma_1, \sigma_2, \ldots, \sigma_n$ of the integers $\{1,2,\ldots,n\}$ satisfying the following two conditions:

For any $1<=i<j<k<l<=n$,

If $\sigma_i+1 = \sigma_l$, then $\sigma_j > \sigma_l \Rightarrow \sigma_k > \sigma_l$ (B_1)

If $\sigma_l+1 = \sigma_i$, then $\sigma_k > \sigma_i \Rightarrow \sigma_j > \sigma_i$ (B_2).

Definition 2.1.2 Baxter number: The Baxter number of order n, B(n), is the number of different Baxter permutations on $\{1,2,\ldots,n\}$.

In [3], Chung et al. proved that B(n) has the nice form (1). This form is derived based on the following lemma:

Lemma 2.1.1: $B(n+1) = \sum_{i,j\ge 0} T_n(i,j), \quad n \ge 0.$ where $T_n(i,j)$ is defined by the following recurrence:

$$T_{n+1}(i+1, j+1) = \sum_{k=1}^{\infty} (T_n(i+k, j) + T_n(i, j+k)), \quad n \ge 0,$$

$$\text{with } T_0(i,j) = \begin{cases} 1 & \text{if } i = j = 0 \\ 0 & \text{otherwise} \end{cases}$$

and $T_n(i,j) = 0$ if $i<0$ or $j<0$ or $i+j>n$.

We now consider the number of mosaic floorplans with n blocks, which is denoted by M(n). We count this number by breaking it down into the numbers of mosaic floorplans with different numbers of T-junctions on the top and right boundaries: Let $F_n(i,j)$ be the number of configurations in a mosaic floorplan of n blocks with i T-junctions on the top boundary and j T-junctions on the right boundary, We have the following equation:

$$M(n) = \sum_{i,j \geq 0} F_n(i,j), \quad n \geq 1. \tag{3.2}$$

where if n= 1, $F_1(0,0) = 1$, and if n > 1, $F_n(0,0) = 0$, since if the number of blocks is larger than 1, there is at least one T-junction on either the top or right boundary. Also we have $F_n(i,j) = 0$ when $i+j \geq n$, since there are at most n-1 T-junctions on both the top and right edges for a floorplan with n blocks. We set $F_n(i,j) = 0$ if i<0 or j<0, because the number of T-junction cannot be negative.

To calculate $F_n(i,j)$, we introduce the following recurrence:

Lemma 2.1.2: The recurrence for $F_n(i,j)$ can be written as:

$$F_{n+1}(i+1, j+1) = \sum_{k=1}^{\infty} (F_n(i+k, j) + F_n(i, j+k)), \quad n \geq 1.$$

From Lemmas 2.1.1 and 2.1.2 we derive equality for the two recurrences.

Lemma 2.1.3: $F_n(i,j) = T_{n-1}(i,j)$ for n ≥1.

Through these two lemmas we get our final theorem on the value of M(n).

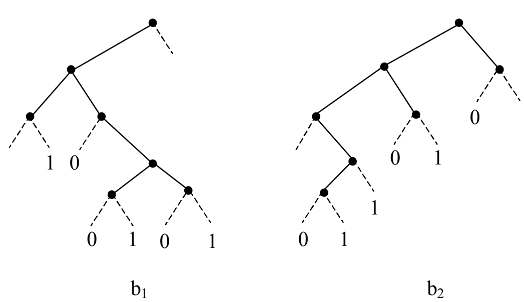
Theorem 2.1.1: The number of combinations of mosaic floorplans with n blocks is equal to the number of Baxter permutations on {1,…, n}, i.e., M(n) = B(n).

2.2 The Twin Binary Tree Representation of a Mosaic Floorplan

Just as we found the relationship between mosaic floorplans and Baxter permutations, it is natural for us to look for an efficient representation of a mosaic floorplan by borrowing the expression of Baxter permutation. In [4], Dulucq introduced a bijection between Baxter permutations on {1,…,n} and twin binary trees TBT_n. In that paper, twin binary trees were originally defined as follows:

Definition 2.2.1 Twin Binary Trees: The set of twin binary trees $TBT_n \subseteq Tree_n \times Tree_n$ is the set

$$TBT_n = \{(b_1,b_2) \mid b_1, b_2 \in Tree_n \text{ and } \Theta(b_1) = \Theta^c(b_2)\}$$



$\Theta(b_1) = 100101$ $\Theta(b_2) = 011010$

Fig. 4 Twin binary trees with 7 nodes

where $Tree_n$ is the set of binary trees with n nodes, and Θ(b) is the labeling of a binary tree b as follows. We begin with an empty sequence, and carry out an in-order walk on the tree. When a node has no left child, we add the letter 0 to the sequence. When it has no right child, we add the letter 1 to the sequence. We omit the first 0 and the last 1 in the resulting sequence. Θ^c is identical to Θ except that letters 0 and 1 are interchanged. Fig. 4 shows an example of twin binary trees with n = 7. Each tree has 7 nodes and $\Theta(b_2)$ is the complement of $\Theta(b_1)$.

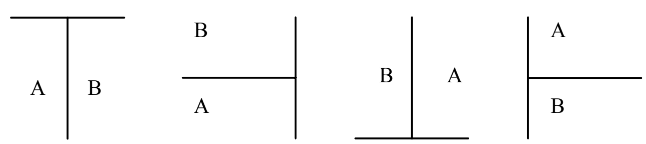
In [4], a sorting algorithm is proposed so that the labels between the pair of twin trees have a one-to-one mapping relation. Consequently, the number of possible twin binary tree combinations is equal to the product of the number of string permutations on n labels, n!, and the possible numbers of twin tree pairs.

To describe the twin-binary representation of a floorplan we need to define some terminology concerning mosaic floorplans:

Definition 2.2.2 *0° T-junction, 90° T-junction, 180° T-junction and 270° T-junction:* In a mosaic floorplan, except for the four corners of the boundary rectangle, a T-shaped intersection has four different orientations [Fig. 5]. We call these four kinds of intersections an *0° T-junction, 90° T-junction, 180° T-junction and 270° T-junction,* respectively.

Definition 2.2.3 *C^+-neighbor:* Given a mosaic floorplan, assume that block A is not the up-right corner block of the chip. The T-junction at the up-right corner of A is either a 0° T-junction or a 270° T-junction. Let B be the block adjacent to A by the non-crossing segment at the corner of that T-junction. B is called the *C^+-neighbor* of A [Fig. 5 a, b].

Definition 2.2.4 *C^--neighbor:* Given a mosaic floorplan, assume that block A is not the bottom-left corner block of the chip. The T-junction at the bottom-left corner of A is either a 90° T-junction or a 180° T-junction. Let B be the block adjacent to A by the non-crossing segment at the corner of that T-junction. B is called the *C^--neighbor* of A [Fig. 5 c, d].



a. 0° T-junction b. 270° T-junction c. 180° T-junction d. 90° T-junction

Fig. 5 Four different type of T-junctions

According to the definition of a C^+-neighbor and a C^--neighbor, we have the following lemma.

Lemma 2.2.1 Except for the up-right corner block of the floorplan, each block of the floorplan has exactly one C^+-neighbor. Except for the bottom-left corner block of the floorplan, each block of the floorplan has exactly one C^--neighbor.

If we connect each node to its C^+-neighbor, we can construct a tree as follows. The root of the tree is the up-right corner block of the floorplan. Similarly, if we connect each block to its C^--neighbor we also can get a tree and the root is the bottom-left corner block. Based on that, the following algorithms generate the twin binary tree representation of a mosaic floorplan.

Algorithm MFTB //Mosaic Floorplan to Twin Binary Tree

1. Initialize two sets of nodes V^+, V^- and two sets of edges E^+, E^- to empty sets.
2. Let $V^+ = V^- = \{i \mid$ there is a block labeled i in the floorplan$\}$

3. For each block i:

3.1 If i is not the up-right corner block of the floorplan, then

{ get C^+-neighbor j of block i and add (j, i) to edge set E^+

If block j is on the right of block i, set i be the left child of j

else set i be the right child of j. }

3.2 If i is not the bottom-left corner block of the floorplan, then

{ get C^--neighbor j of block i and add (j, i) to edge set E^-

If block j is on the left of block i, set i be the right child of j,

else set i be the left child of j. }

4. Let $\tau_+ = (V^+, E^+)$ $\tau_- = (V^-, E^-)$.

5. return (τ_+, τ_-).

Lemma 2.2.2: The complexity of Algorithm MFTB is O(*n*).

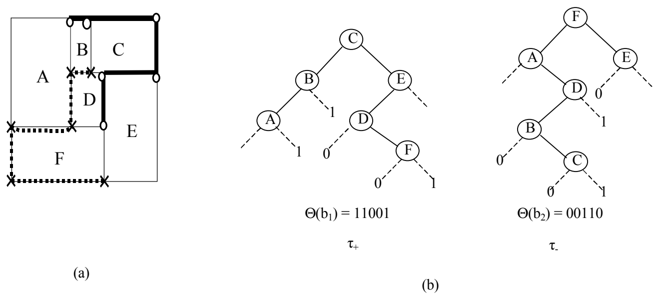


Fig. 6 Twin Binary Tree representation of a mosaic floorplan

Theorem 2.2.1: The graph pair (τ_+, τ_-) created by Algorithm MFTB is a pair of twin binary trees TBT_n

Fig. 6 shows an example of the twin binary tree representation of a mosaic floorplan. In Fig. 6(a), a mosaic floorplan that is the same as the one in Fig. 2 is redrawn and in Fig. 6(b), we present the binary tree representation, (τ_+, τ_-), for the floorplan in Fig. 6(a). To make the derivation of the twin binary trees clear, in Fig. 6(a), we mark the upper-right corner of each block with a circle. These circles correspond to the nodes in τ_+. We mark the lower-left corner of each block with a cross. These crosses correspond to the nodes in τ_-. The line segments between adjacent circles are marked with solid bold lines, which correspond to the edges in τ_+. The line segments between adjacent crosses are marked with dashed bold lines, which correspond to the edges in τ_-.

Moreover, there is one-to-one mapping between the representation of the pair (τ_+ , τ_-) and the mosaic floorplan. To each mosaic floorplan, we can find a single representation of (τ_+ , τ_-), and conversely. This property makes the twin binary tree representation non-redundant in its representation space.

Theorem 2.2.2: Given a mosaic floorplan, there exists a unique twin binary tree representing the floorplan. Likewise, a twin binary tree represents a unique floorplan.

2.3 The exact number of slicing floorplan configurations

Traditionally, a slicing floorplan was represented by a binary tree, which can be obtained by recursively cutting a rectangle into exactly two parts by either a vertical or a horizontal line. This representation is intuitive and simple to implement. However, there is redundancy in the representation. For example, the floorplan in Fig. 7 can be mapped to two different binary trees.

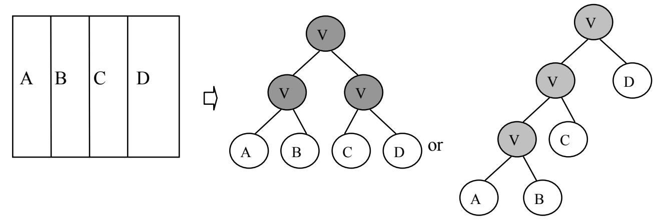


Fig. 7 Example of redundancy in a slicing tree representation

In order to find the exact number of different slicing floorplans of n blocks, we develop a new representation called a slicing ordered tree, and set up a one-to-one correspondence between all possible slicing floorplans and different slicing-ordered trees. We define a slicing ordered tree as follows:

Definition 2.3.1 *Slicing Ordered Tree:* A tree with the following properties is called a s*licing ordered tree:*

1. The tree is a rooted ordered tree with n labeled leaves.
2. Each internal node has at least two children.
3. Each internal node has a label either 'V' or 'H'.
4. Each leaf node has a distinct label.
5. The label of an internal node must be different from its parent's label.

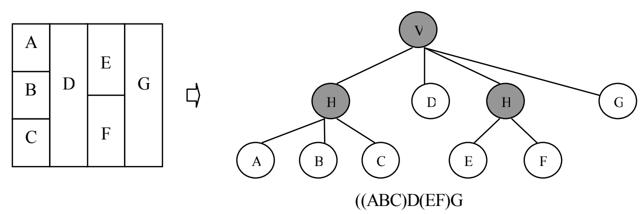


Fig. 8 Slicing Ordered Tree representation of a slicing floorplan

Similar to the traditional binary tree representation, we construct a Slicing-Orderded Tree by recursively cutting a rectangle either vertically or horizontally. The difference is that a Slicing-Ordered Tree is not necessarily a binary tree. Therefore, in each cutting step, we can partition a region into more than two strips either vertically or horizontally. For example, in Fig.8, the root of the tree is labeled V with four children. Correspondingly, the chip is divided into four vertical strips. The most left child of the root is then cut horizontally partitioned into 3 blocks.

Note that slicing floorplans belong to the set of mosaic floorplans. According to property 3 of mosaic floorplans, there is no degenerate case. Using property 4 of a slicing ordered tree, we can prove the-one to-one mapping between slicing ordered trees and slicing floorplans.

Theorem 2.3.1: Given a slicing floorplan, there exists a unique slicing ordered tree representing the floorplan. Likewise, a slicing ordered tree represents a unique slicing floorplan.

Because we can represent an unlabeled slicing-ordered tree by inserting brackets into a list with n characters, as we do in Fig.8, we can get the number of unlabeled slicing-ordered trees with n leaves by counting the combinations of different ways to insert the brackets.

These numbers are known as Schröder numbers or Super-Catalan Numbers: A_n [5]

$$A_0 = 1; A_1 = 1;$$
$$A_n = (3(2n-3)A_{n-1} - (n-3)A_{n-2})/n$$

Theorem 2.3.2: The number of slicing floorplans with n > 1 blocks is twice the Schröder number A_n .

3. RELATIONSHIPS BETWEEN AND REDUNDANCIES OF REPRESENTATIONS

In this section, we describe the relations between several representations and discuss their redundancy.

3.1 Relationships between representations

To demonstrate the relationship between the representations, we use three examples: a slicing floorplan [Fig. 9], a mosaic floorplan [Fig. 10] and a general floorplan [Fig. 11].

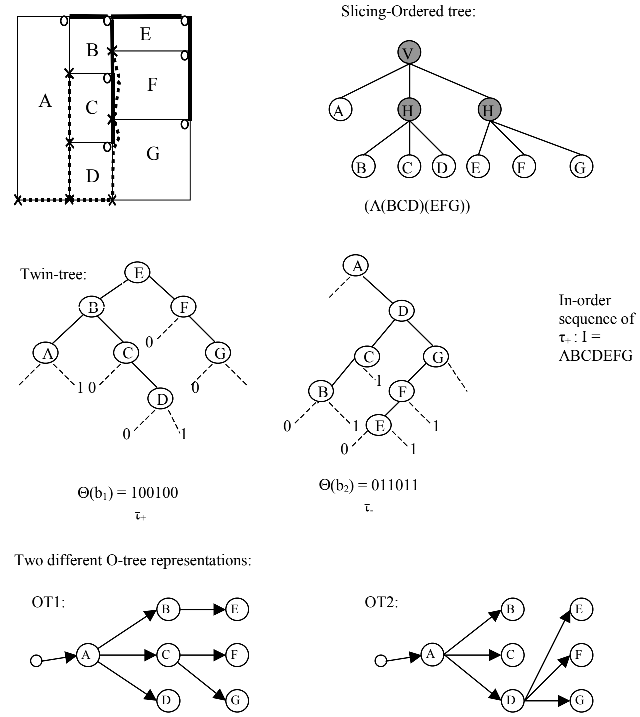
The slicing floorplan in Fig. 9 is first dissected into three strips vertically: the left strip contains A, the middle strip contains blocks B, C and D, and the right strip contains blocks E, F and G. Second, the middle and the right strips are cut horizontally into 3 pieces. The slicing ordered tree representation is shown in Fig. 9. In Fig. 9, we also include the twin binary tree representation. The up-right corner of each block is marked with a circle and the bottom-left corner is marked with a cross. The up-right corners are connected to their C^+-neighbors with solid lines. These solid lines and circles form the τ_+ of the twin binary trees. Similarly the bottom-left corners are connected to their C^- neighbors by dashed lines. These dash lines and crosses form the τ_- of the twin binary trees.

Given a floorplan and the dimensions of blocks, we can find a sequence pair SP=(Γ_+,Γ_+) to represent the floorplan [8]. The Γ_+ of the SP is the order of the blocks from up-left to the right-bottom, and Γ_- is the order of the blocks from bottom-left to up-right. Thus, many SP's may correspond to the same floorplan. For example, in Fig. 9, block F can either be above block D or to the right of block D. Thus, we have different SP representations (SP1, SP2, SP3) [Fig. 9]. With the given block dimensions, the three SP's (SP1, SP2, SP3) produce the same floorplan.

For the floorplan in Fig. 9, there are quite a few different O-tree representations. For example, block F is right adjacent to blocks B and C; thus, node F can be the child of either B or C in an O-tree representation. Two different O-tree representations are shown in Fig. 9. An O-tree can be mapped to a binary tree by converting the sibling relations to left child branches and converting the descendent relations to right child branches of a binary tree [2]. In Fig. 9, O-tree OT2 and the τ_- of the twin binary trees are identical after the tree conversion.

We traverse τ_+ of the twin binary trees in Fig. 9 in a depth-first order and get the in-order sequence of τ_+: I = ABCDEF. This sequence is the same as the first sequence of SP_2. The Corner Block list representation, CB=(S,L,T), is given at the bottom of Fig. 9. We note that the block sequence S=ADCBGFE of CB is identical to the second sequence of SP3.

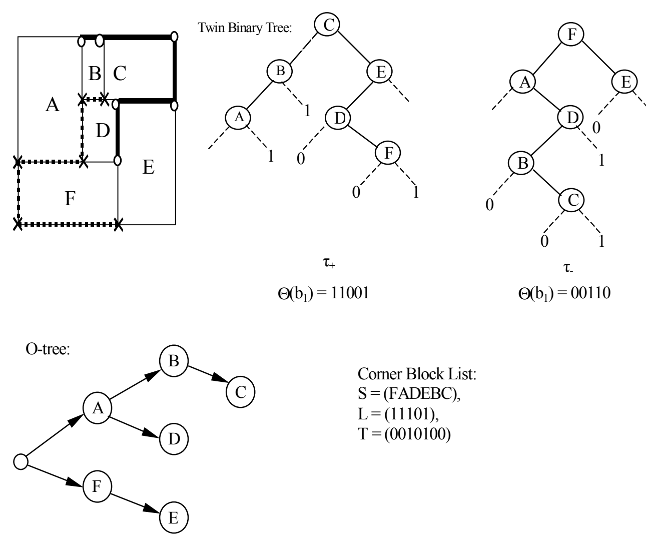
Fig. 10 describes an example of a mosaic floorplan. A slicing-ordered tree is not available to represent this kind of floorplan. We illustrate the other four representations. The twin binary trees are marked by circles and crosses as shown in Fig. 9. Two SP's, SP1 and SP2, out of many possible choices are described in the figure.



One SP representation of this floorplan is $SP_1=(\Gamma_+, \Gamma_-)$ = (ABECFDG, ADCGBFE). Also the sequence pair $SP_2=(\Gamma_+, \Gamma_-)$ = (ABCDEFG, ADCGBFE) and $SP_3=(\Gamma_+, \Gamma_-)$ = (ABCDEFG, ADCBGFE) represent the same floorplan.

CB: S = (ADCBGFE), L = (100100), T=(00011000)

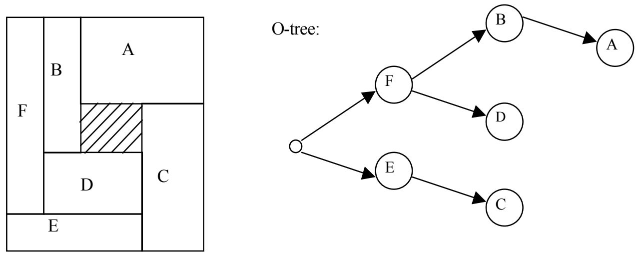
Fig. 9: Different representations for a slicing floorplan



Sequence Pair: $SP_1 = (\Gamma_+, \Gamma_-)$ = (ABCDFE, FADEBC). Also $SP_2 = (\Gamma_+, \Gamma_-)$ = (ABCDFE, FADBEC) refers to the same floorplanning.

Fig 10: Mosaic floorplan and its different representations

The in-order traversal of the τ_+ in the twin binary tree representation produces the sequence I = ABCDFE, which is same as the first sequence of SP1 and SP2. In Fig. 10, an O-tree representation is also



Sequence Pair: SP1=(Γ_+, Γ_-) = (FBADEC, EFDBCA),
Also, SP2=(Γ_+, Γ_-) = (FBADEC, EDFBCA)

Fig. 11: A general floorplan with its O-tree and SP representations

given. Its binary tree representation is identical to the τ_- of the twin binary trees after tree conversion. The CB representation is beside the O-tree representation in Fig. 10. Its block list S=FADEBC is same as the second sequence of SP1.

Fig. 11 illustrates a general floorplan. Only the O-tree and the Sequence Pair are capable of representing a general floorplan. The O-tree and SP representations are shown in the figure.

According to the observations in the above examples, we prove the following lemmas.

Lemma 3.1.1: Given a mosaic floorplan and its corresponding twin binary trees (τ_+, τ_-), there exists a sequence pair SP corresponding to the same floorplan such that the first sequence of SP is the same as the sequence of a depth-first traversal of τ_+.

Lemma 3.1.2: Given a mosaic floorplan and its corresponding corner block list CB=(S, L, T), there exists a sequence pair SP corresponding to the same floorplan such that the second sequence of SP is same as the sequence S of the corner block list.

Lemma 3.1.3: Given a mosaic floorplan and its corresponding twin binary trees TBT (τ_+, τ_-), there exists an O-tree corresponding to the same floorplan such that τ_- is identical to the O-tree after the tree conversion from a binary tree to an ordered tree.

3.2 Redundancy in representations

Redundancy means that there are more than one representations that represent one floorplan, or there is some representation that cannot be mapped to a floorplan. The redundancy in representation can waste steps in various search procedures.

For a corner block list, the list may not correspond to a floorplan. This is because of the constraints of the list T. In [7], a rule for T is given - in any prefix of T, the number of '1's is no more than the number of '0's. This rule eliminates most invalid representations but still cannot guarantee that every CB represents a legal floorplan. For example, S=(ABC), L=(111), T=(0100) is a CB and obeys that rule but there is no floorplan corresponding to it.

In SP, two sequences Γ_+ and Γ_- sort all the blocks from up-left to bottom-right and from bottom-left to up-right. When the relative positions of two blocks is both up and right, their relative position in Γ_+ have multiple choices as shown in Fig. 9. Similarly, if the relative position of two blocks is both down and right, their relative positions in Γ_- have multiple choices. This redundant representation causes a one-to-many mapping from a floorplan to the representations.

In twin binary tree, there is no redundancy in representing mosaic floorplan, since the mapping is exactly one-to-one. This characteristic justifies its conciseness.

According to Lemma 3.1.3, an O-tree provides the information of one of twin binary trees only. Different choices of the other twin of the twin binary trees represent different mosaic floorplans. On the other hand, the O-tree is claimed to cover all possible compact floorplans [6]. This is because O-trees also rely on the dimensions of the blocks. The variations of the block dimensions make up for the possible choices contributed by the other twin of the twin tree pair. By amortizing the block dimensions, O-trees require a very small number of combinations to represent a floorplan.

4. CONCLUSIONS

We prove that the numbers of mosaic floorplan and slicing floorplan configurations are Baxter numbers and Schröder numbers, respectively. Two tree structures, twin binary trees and ordered slicing trees, are proposed to represent mosaic floorplans and slicing floorplans. These two representations are concise in the sense that the transformation between the representation and the floorplan is a one-to-one mapping. We also identify the relations of several representations, where the proposed twin binary trees serve as a bridge for the relation.

5. ACKNOWLEDGEMENT

The authors wish to thank the referees for many valuable comments. This work was supported in part under grants from NSF project number MIP-9987678 and California MICRO program.

REFERENCES

[1] G. Baxter, "On Fixed Points of the Composite of Commuting functions", in Proc. Amer. Math. Soc., Vol. 15, 1964, pp.851-855.

[2] Y. Chang, Y. Chang, G. Wu, and S. Wu, "B*-Trees: A New Representation for Non-Slicing Floorplans", in Proc. 37th DAC, 2000, pp. 458-463.

[3] F.R.K. Chung, R.L.Graham, V. E. Hoggatt, Jr. and M. Kleiman, "The Number of Baxter Permutations," in Journal of Combinatorial Theory, Series A, Vol. 24, No. 3, May 1978, pp. 382-394.

[4] S. Dulucq and O. Guibert, "Baxter permutations," in Discrete Mathematics, Vol. 180, 1998, pp. 143-156. (twin Binary Tree)

[5] I. M. H. Etherington, "Some Problems of Non-Associative Combinations", in The Edinburgh Mathematical Notes, No. 32, 1940, pp. i-vi

[6] P. Guo, C. K. Cheng, and T. Yoshimura, "An O-Tree Representation of Non-Slicing Floorplan and Its Applications," in Proc. 36th DAC, 1999, pp. 268-273.

[7] X. Hong, et al, "Conner Block List: An Effective and Efficient Topological Representation of Non-Slicing Floorplan," in Proceeding of ICCAD-2000, 2000, pp. 8-12

[8] H. Murata, et al, "Rectangle-packing-based Module Placement", in Proc. of International Conference on Computer Aided Design, 1995, pp. 472-479.

[9] S. Nakatake, et al. "Module Packing Based on the BSG-Structure and IC Layout Applications", in IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 17, No. 6, June 1998, pp. 519-530.

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

[11] X. Tang, M. Wong "FAST-SP A Fast Algorithm for Block Placement based on Sequence Pair", in ASP-DAC 2001.