The Quarter-State Sequence (Q-sequence) to Represent the Floorplan and Applications to Layout Optimization

Keishi Sakanushi and Yoji Kajitani

Tokyo Institute of Technology
Department of Communications and Integrated Systems
2-12-1 Ookayama,Meguro-ku,Tokyo 152-8552,Japan
E-mail: {keishi,kajitani}@ss.titech.ac.jp

Abstract

A new data structure "Q-sequence" for representing a floorplan of n rooms is proposed. The Q-sequence is a concatenation of room names and two kinds of positional symbols, totally of length 3n. It is shown that encoding of a given floorplan and decoding to a floorplan are both possible in linear time of n. An exact counting formula of distinct floorplans is given. Numerical estimation shows that the number is only slightly larger than that of slicing structure, and far smaller than $(n!)^2$ which is the size of the packing solution space by the Sequence-Pair.

keyword *floorplan, Q-sequence, count, 4-direction, topological packing*

1 Introduction

Floorplanning in VLSI layout design is a divide-conquer strategy to the layout design of a huge scale circuit. Given a circuit consisting of n functional sub-circuits (modules) to be placed on a rectangular area (chip), a floorplan is a dissection of the chip into n rectangles (rooms), each to house one module. Optimization is hierarchical such that each module shall be realized, or chosen from IP's, optimally inside the room, and then modules are interconnected global-optimally via terminals. All these optimizations are controlled by the floorplan.

Since floorplanning is a top down strategy, it is so much indirect to identify how and what features of the floorplan affect the performance. Furthermore, the evaluation function of the circuit with respect to layout is very complicated these days, even to the level of adopting a look-up-table[2]. In such environments, a constructive way is considered not prospective except for providing an initial solution, and the stochastic sampling approach is taking its place. It searches the solution space of floorplans one after another, evaluating each assuming a rough routing, as long as the computation resource allows, to output the best so far.

The base that makes such a search being effective is in the representation of the floorplan. It must be smart enough that any floorplan can be encoded, that encoding and decoding are both easy, and that a systematic generation of all representations is possible. The purpose of this paper is to introduce an idea.

So far, an elegant coding system has been known only for the floorplans with the slicing structure[3]. For the general structured floorplan, a study is found in [8] where a coding system for general floorplans is proposed. Its purpose is to estimate the number of distinct floorplans. But the result was a rough upper bound. Moreover, the room names are implicit in the code so that it is not suitable for layout design. However, it motivated our work, though the idea is very much simplified and rooms become explicit.

It must be noted about the fact that the packing algorithm can be an alternative for floorplanning. Letting the input be n rectangles, the algorithm will output a non-overlapping placement of them in the chip of as small area. If we regard each rectangle a room, we can have a floorplan with at least n rooms. Often the packing contains a space that cannot be covered by the rectangles whatever their sizes are, which we don't admit as a floorplan. Still recent advances in packing algorithms, which was initiated by the general structure topological packing BSG [6] and Sequece-Pair [4], or partly topological O-tree [9, 10], attempts us to apply them for floorplanning. In fact, some efforts based on the Sequence-Pair have been reported[7], where a check routine is added to reject inadequate sequence-pair, by which the computation time to get one floorplan amounts to $O(n^2)$.

We propose a new coding system named the Quarter-state sequence (abbreviated as Q-sequence). It is a data structure of length $3n$, consisting of room names and $2n$ positional symbols that describe the states of rooms. Since it is constructed as a concatenation of the local state of each room, encoding and decoding are possible in linear time and operations such as removing a room or inserting can be executed in a constant time. A byproduct is an exact formula to count all distinct floorplans, which has been a long-standing problem in combinatorial mathematics.

We start in Section 2 with a note that there are two definitions of the "floorplan", which will make clear our standpoint. In Section 3 and 4, we explain the coding and decoding. In Section 5, we derive a counting formula. Section 6 summarizes our contributions.

2 Preliminary Discussions

First of all, the criterion of floorplan must be clarified. Line segments that dissect the inside area of the chip and those that enclose the whole area are called *inside-*

segs or *wall-segs*, respectively. When it is not necessary to distinguish them, they are called simply *segs*. Segs do not cross each other and an inside-seg ends at another orthogonal seg in T-form.

The relation between a seg and a room that the seg is the left wall (right-wall, above-wall, or below-wall) of the room is called the *room-seg* relation. Two floorplans are said equivalent (with respect to room-seg relation), if it is possible to label the rooms and segs in such a way that the room-seg relations are identical. This is the floorplan we are considering in this paper.

A different but probably more popular criterion of the floorplan is defined as follows. The relation of two rooms such that they share the same part of a seg is called the *room-room* relation. If there is a labeling of rooms such that the room-room relations are identical, these floorplans are said equivalent (with respect to room-room relation). See a difference in a simple example in Fig. 1. (There have been many studies in eighties about the room-room floorplans, and now considered fixed since the floorplan is completely characterized in terms of *complex triangle* and its feasibility is verified in $O(n)$ time. Literature [1] is referred for overview.)

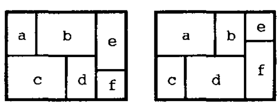

Figure 1: Two floorplans equivalent w.r.t.the room-seg relations but not w.r.t. the room-room relation.

It is an essential question which of the room-seg and room-room relations are more significant in VLSI layout design, especially from the point of routing. Apparently, two rooms in a room-room relation are preferable because they are easily interconnected by short wires or directly. But this merit should not be expected too much since it is observed only when the pitch of pin positions of two modules matches exactly, which is very rare unless so designed. Then, we need a channel between them anyway. Furthermore, the criterion of feasible floorplans is too strict for the designer's practical request to be accepted. Thus we must be satisfied if two modules face the same channel, if they are to be strongly interconnected. This is materialized in practice in the layout style of *standard array* where modules are fabricated to have the same height and laid out in lines so that as many modules face the same channel. In this layout, a further significance is if two modules are on the same side or not of a channel. This influences considerably in compaction if we think of interaction with other routings. These discussions support the claim that the floorplan based on the room-seg relation is significant in VLSI floorplanning.

3 Coding of a floorplan

Assume a floorplan of n rooms such that every room has its name. Of symmetric discussions with respect to the system of *4-direction* (left, right, top, bottom), we go with one of them without loss of generality.

A room is called the *tail room* if it lies at the right-below corner. Take a room r that is not a tail room. At its right-below corner, the vertical or horizontal seg ends at the other in T-form. In each case, the seg is called the *prime seg* of r. See Fig. 2.

If the prime seg is vertical (horizontal), the rooms that touch the seg from the right (below) are called the *associated* rooms. Then, the *Q-state* of room r is defined as the sequence of letter r followed by labels of $\mathcal{R}$ ($\mathcal{B}$) of the number of associated rooms. The room that lies topmost (leftest) of the associated rooms is called the *next room* of r. See Fig.2.

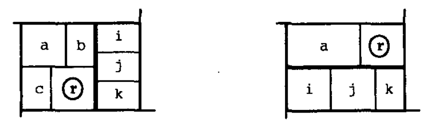

Figure 2: With respect to room r: prime seg (bold line), associated rooms i, j, k, next room i, and Q-state: $(r\mathcal{RRR})$(left) and $(r\mathcal{BBB})$(right).

The *subQ-sequence* is the concatenation of Q-states of all rooms along the order of

> left-top room comes first, then the next room, and its next room so on, to end by the tail room.

For example, the subQ-sequence of the floorplan in Fig.1 is $(a\mathcal{R}b\mathcal{BB}c\mathcal{R}d\mathcal{RR}e\mathcal{B}f)$.

A room is called the *inside room* if it does not touch the left-wall nor above-wall of the chip. Take an inside room r. There is a preceding room whose prime seg is a horizontal seg that associates $\mathcal{B}$ pointing room r. Also there is a different room whose prime seg is a vertical seg that associates $\mathcal{R}$ pointing room r. In the above example, inside room d is predicted by one of $\mathcal{B}$ in $(\cdots b\mathcal{BB}\cdots d)$ and one of $\mathcal{R}$ in $(\cdots c\mathcal{R}d\cdots)$. To make this kind of property hold for every room, we add a pre-sequence X to the subQ-sequence Q' in a fashion that (X Q') where X is the concatenation of $\mathcal{R}$'s and $\mathcal{B}$'s of the numbers of rooms that touch the left-wall or above-wall of the chip, respectively. The resultant is unique and called the *Q-sequence*.

A more complex example is shown in Fig.3, where rooms are labeled with integers to represent the inverse order of appearance. The reason of this labeling will be understood later.

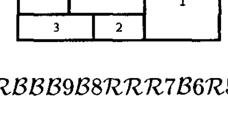

Figure 3: $(\mathcal{RRRBBB}9\mathcal{B}8\mathcal{RRR}7\mathcal{B}6\mathcal{R}5\mathcal{BB}4\mathcal{BB}3\mathcal{R}2\mathcal{R}1)$

4 Decoding of a Q-sequence

The above-mentioned property is characterized by mathematical terms.

– **Procedure: Parent (Q to $\mathcal{R}$Q)** –

1. Delete all the $\mathcal{B}$'s. (The resultant is called the $\mathcal{R}Q$-*sequence.*)
2. Replace every $\mathcal{R}$ with an open parenthesis "(".
3. Replace every room name with a close parenthesis ")".□

Similarly, procedure **Parent (Q to $\mathcal{B}$Q)** is defined by exchanging $\mathcal{B}$ and $\mathcal{R}$. In Fig.4, procedures are shown in one configuration, with parenthesis correspondences.

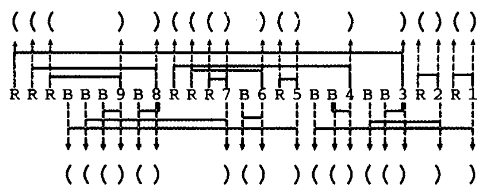

Figure 4: Q-sequence and procedures Parent(Q to $\mathcal{R}$Q) and Parent(Q to $\mathcal{B}$Q) for a floorplan given in Fig.3.

Theorem 1 *A sequence consisting of n $\mathcal{R}$'s and n $\mathcal{B}$'s and n room names is the Q-sequence if and only if following two conditions are satisfied.*

> **Single:** *The subsequence between two rooms contains one or more single labels of $\mathcal{R}$ or $\mathcal{B}$, and*
>
> **Parent:** *The parenthesis system by $\mathcal{R}Q$-sequence or $\mathcal{B}Q$-sequence is consistent.*

5 Number of Floorplans

5.1 Exact number

Apart from applications, the strongest interest in combinatorial theory is to give a counting formula of distinct floorplans for given n. We present one based on the Q-Sequence.

Remember that we have set the rule that room names are labeled in reverse order of their appearance in the Q-sequence. For such a Q-sequence, we define two functions $\mathcal{R}$-function $N_{\mathcal{R}}(x)$ and $\mathcal{B}$-function $N_{\mathcal{B}}(x)$ for $x = 1, 2, \ldots, n$. $N_{\mathcal{R}}(x)$ is the number of letters $\mathcal{R}$ in the right of x. $N_{\mathcal{B}}(x)$ is analogously defined. For example in Fig.3, they are as follows.

$$N_{\mathcal{R}}(1) = 0, N_{\mathcal{R}}(2) = 1, N_{\mathcal{R}}(3) = 2, \cdots, N_{\mathcal{R}}(9) = 6$$

$$N_{\mathcal{B}}(1) = 0, N_{\mathcal{B}}(2) = 0, N_{\mathcal{B}}(3) = 0, \cdots, N_{\mathcal{B}}(9) = 6.$$

Note that the pair completely characterizes the Q-sequence. Properties Single and Parent in the theorem are translated by the pair $(N_{\mathcal{R}}(x), N_{\mathcal{B}}(x))$ as follows.

Single' One of $N_{\mathcal{R}}(x) - N_{\mathcal{R}}(x-1)$ and $N_{\mathcal{B}}(x) - N_{\mathcal{B}}(x-1)$ is positive and the other zero.

Parent' $N_{\mathcal{R}}(x) < x$ and $N_{\mathcal{B}}(x) < x$.

The function pair is defined on a given Q-sequence. It is used here to classify the Q-Sequences. Let $C(n, r, b)$ be the set of all Q-Sequences satisfying $N_{\mathcal{R}}(n) = r$ and $N_{\mathcal{B}}(n) = b$. Let $|C(n, r, b)|$ denote the number of distinct elements of the set. Note that b and r range from 0 to $n-1$. In fact, $|C(n, 0, b)| = |C(n, r, n-1)| = 1$ and $|C(n, n-1, b)| = |C(n, r, 0)| = 1$ for any n. Then the number $F(n)$ of distinct Q-Sequences is given by

$$F(n) = \sum_{r=0}^{n-1}\sum_{b=0}^{n-1} |C(n, r, b)| \qquad (1)$$

Furthermore, by property Single, $C(n, r, b)$ is the union of $C(n-1, r', b)$ over $r' < r$ or the union of $C(n-1, r, b')$ over $b' < b$. If $n = 1$, the Q-Sequence is uniquely ($\mathcal{R}\mathcal{B}1$). Hence $|C(1, 0, 0)| = 1$. Thus, we have a recursive formula.

$$|C(n,r,b)| = \begin{cases} 1 & : n = 1, r = b = 0 \\ 0 & : r, b \geq n \\ \sum_{r'=0}^{r-1} |C(n-1, r', b)| & \\ + \sum_{b'=0}^{b-1} |C(n-1, r, b')| & : otherwise \end{cases} \qquad (2)$$

By the system of equations (1) and (2), we can enumerate exact $F(n)$.

5.2 Upper bound

Consider a Q-Sequence. By property Parent, the variety of the Q-sequence is known to be Catalan number of n, $Catalan(n) \sim 2^{2n}/\sqrt{\pi n}$.

Given an $\mathcal{R}$Q-sequence, consider the problem to reconstruct the corresponding Q-sequences. The consecutive intervals where letter $\mathcal{B}$ can be inserted are before room n and between two rooms. However, the interval that has letter $\mathcal{R}$ denies letter $\mathcal{B}$ (by property Single), and every interval without letter $\mathcal{R}$ must have at least one $\mathcal{B}$ (by property Parent). It is also true that any insertion of n $\mathcal{B}$'s to satisfy the above properties gives a unique Q-sequence. Therefore, for one fixed $\mathcal{R}$Q-sequence, the variety of corresponding Q-sequences is the variety of distributions of n $\mathcal{B}$'s into $n - m$ ordered nonempty classes, where m is the number of intervals which letter $\mathcal{R}$ occupies. It is ${}_{n-1}C_{n-m}$, which is bounded by ${}_{n-1}C_{\frac{n-1}{2}}$.

On the other hand, the variety of room names is $n!$. Since different $\mathcal{R}$Q-sequences never generate identical Q-sequences,

$$\begin{aligned} F(n) &\leq Catalan(n) \times {}_{n-1}C_{\frac{n-1}{2}} \times n! \\ &\leq Catalan(n) \times 2^{(n-1)} \times n! \\ &\leq 2^{2n} \times 2^{(n-1)} \times n! = 2^{(3n-1)} \times n! \end{aligned}$$

5.3 Observations

The number of distinct binary trees of n leaves with two kinds of inner node labels is $BI(n) = Catalan \times 2^{(n-1)} \times n!$. Since any slicing floorplan is represented by this class of binary trees, $BI(n)$ is also an upper bound of the number of slicing floorplans, very restricted floorplans. Therefore, the fact that $BI(n)$ is also an upper-bound of the number of general floorplans is significant.

We denote by $SL(n)$ the number of distinct slicing structure floorplans. A self-evident inequality is $SL(n) \leq F(n) \leq BI(n)$. Our interest is to compare the difference between $F(n)$ and $SL(n)$. The former is numerically calculated by equations (1) and (2). The latter is to count the number of *normalized binary trees* defined in [3]. Normalizing by $n!$, the comparison up to $n = 20$ is shown in Fig. 5. In the graph, function $n!$ is also included for reference to compare the size $SP(n) = (n!)^2$ of the packing solution space by the Sequence-Pair.

From these observtions, we see that no significant reduction of the solution space is attained if we restrict the floorplan to the slicing structure. With almost the same computaion labor, we could search the space of general structure floorplans by the Q-sequence.

It is not prospective to apply a packing algorithm by the Sequece-Pair since its solution space is too large.

The upper-bound given in [10] is $Catalan(n) \times (n!)^2$. It is far worse compared our upper-bound $Catalan(n) \times 2^{(n-1)} \times n!$.

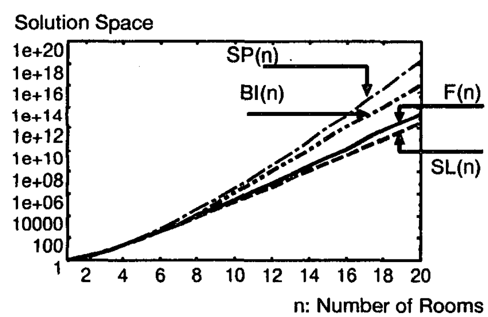

Figure 5: Comparison of sizes of solution spaces.

6 Concluding Remarks

For the space, we did not mention our results about basics properties of the Q-sequence. Examples are:

$ Given a Q-sequence, the vertical (horizontal) constraint graph of the corresponding floorplan can be constructed directly from the sequence.

$ Deletion or insertion of a room can be done in a constant time,

$ Defining the move between solutions by the pair of deletion and insertion, a stochastic algorithm that searches floorplans is implemented. An application is in packing of n soft modules. For each floorplan, we assign modules and apply a very fast soft-module compaction algorithm[5]. The quality by experiments was satisfactory.

$ Boundary check, i.e. listing the rooms that are adjacent to a specified seg is possible in a linear time.

It is believed that a further study will make this data structure most useful in VLSI layout.

Acknowledgement

The authors are grateful for Professor A. Takahashi, Tokuou Institute of Technology and many graduate students in our laboratory, for their valuable discussions. This work is part of a project of CAD21 at Tokyo Inst. of Tech. and is supported in a Research and Development Aid of Support Center for Advanced Telecommunications Technology Research.

References

[1] Y-T. Lai and M. Leinwand, "A theory of Rectangular Dual Graphs," Algorithmica, Vol.5, pp.467–483, 1990.

[2] J. Lilis and P. Buch, "Table-Lookup Methods for Improved Performance-Driven Routing," Proc. 35th DAC, pp.368–373, 1998.

[3] D. F. Wong, and C. L. liu, "A New Algorithm for Floorplan Designs," DAC'23, pp.101–107, Jul., 1986.

[4] H. Murata, K. Fujiyoshi, S. Nakatake, and Y. Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair," TCAD, Vol.15, No12, pp.1518–1524, Dec., 1996.

[5] T. Izumi, A. Takahashi, and Y. Kajitani, "Air-Pressure-Model-Based Fast Algorithms for General Floorplan," APSDAC'98, pp.563–570, Jan., 1998.

[6] S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani, "Module Packing Based on the BSG-Structure and IC Layout Applications," TCAD, Vol.17, No.6, pp.519–530, Jun., 1998.

[7] K. Kiyota and K. Fujiyoshi, "Simulated Annealing Search through General Structure Floorplan using Sequence-Pair," Proceedings of 12th Karuizawa Workshop on Circuits and Systems, pp.457–462, Apr., 1999 (in Japanese).

[8] T. Shimizu, The Rectangular Dissection, Nippon Hyouronsya, Apr., 1999 (in Japanese).

[9] P. N. Guo, C. K. Cheng, and T. Yoshimura, "An O-Tree Representation of Non-Slicing Floorplan and Its Applications," DAC'36, pp.286–291, Jun., 1999.

[10] T. Takahashi, "A New Encoding Scheme for Rectangle Packing Problem," ASPDAC'2000, pp.175–178, Jan., 2000.