

# An Extended Representation of Q-sequence for Optimizing Channel-Adjacency and Routing-Cost

Changwen Zhuang

Keishi Sakanushi

Liyan Jin

Yoji Kajitani

System Development Dept.  
SII EDA Technologies Inc.  
Kitakyushu, Fukuoka 808-0135  
email:changwen@sii.co.jp

Dept. Info. Science & Technol.  
Osaka University  
Osaka 560-8531  
sakanusi@ist.osaka-u.ac.jp

Dept. of Electrical & Electronic Eng.  
Tokyo Inst. of Technol.  
Tokyo 152-8552  
lyjin@lab.ss.titech.ac.jp

Dept. of Info. & Media  
Univ. of Kitakyushu  
Kitakyushu, Fukuoka 808-0135  
kajitani@env.kitakyu-u.ac.jp

**Abstract—** This paper proposes a topological representation for general floorplan, called the *H-sequence*, which can check channel-adjacency and boundary-adjacency in a constant time. Moreover, we define *Routing-cost* for the placement to measure its routing difficulty. Experimental results indicate that H-sequence based placement algorithm can optimize routing-cost effectively in a short time.

## I. INTRODUCTION

### A. Background

Floorplanning is to dissect a rectangular area (*chip* hereinafter) by non-crossing horizontal and vertical line segments (*segs*) into the small rectangles (*rooms*). After placing modules into rooms in one-to-one manner and 1-dimensional compaction, we can evaluate the chip's characteristics such as area, wire length, signal-delay critical path, etc. Therefore, like packing-based representation[1, 2, 3], a floorplan representation can be used to represent and optimize the placement, the first and dominant stage of layout[4].

It is a common objective of VLSI floorplanning to maximize the *channel-adjacency* between modules because it directly leads to a easy-routing placement with short wire length. Moreover, in modern circuit design, IP are extensively used. Since IP and other macro units always occupy many layers, then there are many channels in the chip, so channel adjacency become more important. Therefore, instead of directly approaching to placement's multi-objectives, this paper proposes a completely new and fast general floorplan presentation to evaluate the channel-adjacency of modules in the circuit.

While discussing about channel-adjacency, it is necessary to recollect incidence-adjacency. Two rooms are said to be channel-adjacent if they are placed along the same seg, while two rooms are said to be incidence-adjacent if they are channel-adjacent and their borders along the same seg overlap partially. For example, in Fig. 3, room 2 and room 7 are channel-adjacent, but not incidence-adjacent, while room 2 and room 5 are incidence-adjacent.

Although the incidence-adjacency is also a very naive request from the circuit designers and has been studied extensively[5, 6], it has not become popular yet, maybe because there are no fast and practical floorplan representations to support it.

### B. Our Contributions

First, based on the borders where there are pins of nets, *routing-cost* of the placement is defined to reflect the channel-adjacency of modules in the circuit. The basic objective of placement is to minimize the routing-cost.

Secondly, a new floorplan representation named *Half-state sequence*, called *H-sequence* for short, is introduced. It is a string of room names and positional symbols. Any floorplan is uniquely encoded to an H-sequence and any H-sequence is uniquely decoded to a floorplan, both in linear time. **For any pair of two rooms, it takes only a constant time to check if they are channel-adjacent. This is a generalization of the method[7] that handles floorplans under the boundary constraint.**

Thirdly, the trivial but unpractical solution is a floorplan in which all the rooms are along one seg so that every pair of rooms are channel-adjacent. To avoid such solutions, we limit the number of rooms that are adjacent to the same seg.

Finally, our experiments are planned for three optimizations: *routing-cost optimization*, *restricted routing cost optimization*, and *simultaneous area and routing-cost optimization*. The results on five benchmarks show that the algorithm can optimize the area and routing cost for a circuit.

The rest of the paper is organized as follows. Section II defines routing-cost of a placement. Section III describes the H-sequence. Section IV introduces several operations of perturbing the placement. Section V shows experimental results. Section VI concludes the paper.

## II. ROUTING-COST

It is assumed here that pins are located on the *borders* (peripherals) of rooms. The connection requirement is given by the *net-list* (a collection of *nets*). Each net consists of a set of pins that should be connected through wires.

If two modules  $i$  and  $j$  are channel-adjacent, pins of a net facing the common-channel will be connected by wires routed in this channel. Since it is very straightforward to route such nets, floorplanning that renders as many nets to distribute in such a way is a target of floorplanners.

First we give the definition of the *routing-cost* by an example of two modules  $i$  and  $j$  as shown in Fig.1. Here pins

are labeled as  $a, b, c, d, e, f, g$  such that pins of the same label belong to the same net, i.e. to be connected. For simplification, we assume that there are no two pins belonging to the same net on one module. Let four borders of a module be called  $T(op)$ ,  $R(ight)$ ,  $B(ottom)$ , and  $L(eft)$  and letters  $X, Y$  represent  $T, R, B$ , or  $L$ . The number of pairs of pins on  $X$  of  $i$  and  $Y$  of  $j$  with the same label is called the *border-connectivity* denoted by  $BC_{XY}(i, j)$ . Let  $BC(i, j)$  is the set of all border pairs. In Fig.1,  $BC_{TT}(i, j) = 0$ ,  $BC_{TR}(i, j) = 3$ ,  $BC_{TB}(i, j) = 1$ ,  $\dots$ , and  $BC(i, j) = 7$ . In the figure(right), they are represented in a bipartite graph.

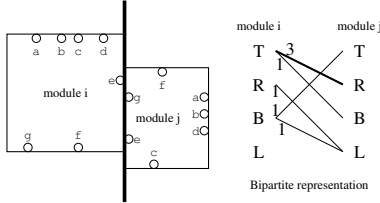


Fig. 1. Border-connectivity and matched-borders: ( $T$  of  $i$ ,  $R$  of  $j$ ) is the matched-border pair

The routing-cost of two modules is based on the border-connectivities of 16 border pairs. However, it is so complicated that we propose a simplification. Given a pair of modules, the pair of borders with maximum border-connectivity is called *matched-border pair*.

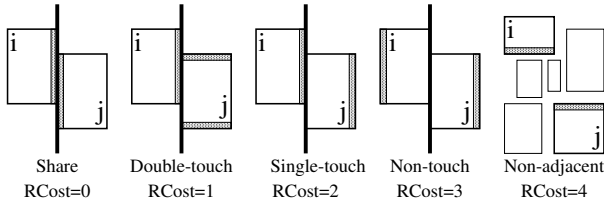


Fig. 2. Five cases of simplified routing-cost

As illustrated in Fig.2, when  $i, j$  are channel-adjacent (common-channel is  $S$ ), 4 cases are named as follows:

- Case (Double-touch) both  $X, Y$  touch  $S$ .
- Case (T-form-touch): one touches  $S$  and the other touches  $S$  in T-form,
- Case (Single-touch): one touches  $S$  but the other does not touch  $S$ ,
- Case (Non-touch) none of  $X, Y$  touches  $S$ .

Then, coefficient  $R_{ij}$  of routing-cost between  $i$  and  $j$  is defined by:

$$\begin{aligned} R_{ij} &= 0 && \text{if (Double-touch)} \\ &= 1 && \text{if (T-form-touch)} \\ &= 2 && \text{if (Single-touch)} \\ &= 3 && \text{if (Non-touch)} \\ &= 4 && \text{if (i, j) is not channel-adjacent.} \end{aligned}$$

Finally, the routing-cost  $R(\pi)$  of a placement  $\pi$  is defined by

$$R(\pi) = \frac{1}{2} \sum_{i,j} R_{ij} \times BC(i, j)$$

### III. HALF-STATE SEQUENCE

#### A. Definition of H-sequence

Let  $s$  be one of  $n + 3$  segs (including 4 borders) in the chip, and  $S(s)$  be the *seg-state* which is obtained by concatenating sequentially four sequences, *right-state*  $\mathcal{R}(s)$ , *left-state*  $\mathcal{L}(s)$ , *bottom-state*  $\mathcal{B}(s)$  and *above-state*  $\mathcal{A}(s)$ . Letters  $\mathcal{R}, \mathcal{L}, \mathcal{B}, \mathcal{A}$  are called the *positional symbols* standing for Right, Left, Bottom, and Above states, respectively.

If seg  $s$  is vertical,  $\mathcal{B}(s)$  and  $\mathcal{A}(s)$  are defined as the empty sequences, while  $\mathcal{R}(s)$  ( $\mathcal{L}(s)$ ) is the sequence of the symbols  $\mathcal{R}$  ( $\mathcal{L}$ ) with suffixes of room names, where the rooms are adjacent to  $s$  and at the right (left) of  $s$ , arranged from the bottom to up. For example, in Fig. 3,  $\mathcal{R}(s_1) = \mathcal{R}_4 \mathcal{R}_2$ .

If seg  $s$  is horizontal,  $\mathcal{R}(s)$  and  $\mathcal{L}(s)$  are defined as empty, while  $\mathcal{B}(s)$  ( $\mathcal{A}(s)$ ) is the sequence of the symbols  $\mathcal{B}$  ( $\mathcal{A}$ ) with suffixes of room names, where the rooms are adjacent to and below (above) $s$ , arranged from the right to left. For example, in Fig. 3,  $\mathcal{B}(s_3) = (\mathcal{B}_7 \mathcal{B}_5 \mathcal{B}_4)$ .

In Fig. 3, the right-, left-, below-, and above-states of  $s_3$  are  $\mathcal{R}(s_3) = \emptyset$ ,  $\mathcal{L}(s_3) = \emptyset$ ,  $\mathcal{B}(s_3) = (\mathcal{B}_7 \mathcal{B}_5 \mathcal{B}_4)$ , and  $\mathcal{A}(s_3) = (\mathcal{A}_3 \mathcal{A}_2)$ , respectively. Then the seg-state  $S(s_3)$  of  $s_3$  is

$$S(s_3) = \mathcal{B}_7 \mathcal{B}_5 \mathcal{B}_4 \mathcal{A}_3 \mathcal{A}_2.$$

If  $s$  is the left(top)border,  $\mathcal{L}(s)$  ( $\mathcal{A}(s)$ ) is empty. Analogous facts hold when  $s$  is right or bottom-border.

Other examples of the seg-states in Fig. 3 are

$$\begin{aligned} S(\text{left-border}) &= \mathcal{R}_6 \mathcal{R}_1, \\ S(\text{top-border}) &= \mathcal{B}_3 \mathcal{B}_2 \mathcal{B}_1, \\ S(s_1) &= \mathcal{R}_4 \mathcal{R}_2 \mathcal{L}_1, \\ S(s_5) &= \mathcal{B}_6 \mathcal{A}_5 \mathcal{A}_4 \mathcal{A}_1, \\ S(\text{right-border}) &= \mathcal{L}_7 \mathcal{L}_3, \\ S(\text{bottom-border}) &= \mathcal{A}_7 \mathcal{A}_6. \end{aligned}$$

The key issue of the H-sequence is *Abe-order* of rooms [8, 9, 10], which can be easily defined according to the definition of the seg-state. Please look at the room  $k$  which is not the right-bottom room in Fig.3 (a) and (b), the seg that ends at the right-bottom corner of the room  $k$  in a T-form is called the *prime seg* of room  $k$  and denoted as  $s_k$ . if  $s_k$  is vertical(horizontal), the room corresponding to the last positional symbol in  $\mathcal{R}(s_k)$  ( $\mathcal{B}(s_k)$ ) is called the *next room* of room  $k$ . For example, The room corresponding to the last  $\mathcal{B}$  positional symbol in  $S(s_3)$  is 4, so 4 is the *next room* of room 3.

The *Abe-order* of rooms are linear ordering of rooms defined as follows. (See the labeling in Fig.3(a).)

Label the left-top room with 1. The *next room* of 1 is labeled with 2. In general, the *next room* of room  $k$  is labeled  $k + 1$ , where  $1 \leq k < n$ .

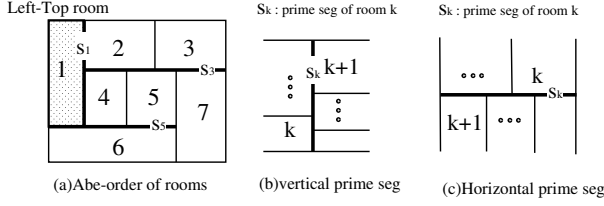


Fig. 3. Prime seg,next room, and Abe-order

It was proved that every room is labeled uniquely and the final room, room  $n$ , is the right-bottom room of the chip [8, 10, 9].

Note that the seg-state is the sequence of positional symbols with decreasing suffixes in each symbol.

Now the  $H$ -sequence of a floorplan can be defined.

$$\begin{aligned} H\text{-sequence} = & S(\text{left-border}) S(\text{top-border}) 1 S(s_1) \\ & 2 S(s_2) 3 S(s_3) \cdots S(s_{n-1}) n S(\text{right-border}) \\ & S(\text{bottom-border}). \end{aligned}$$

The integers in the H-sequence are called *room labels*.

In Fig.3(a), the H-sequence of the floorplan,  $H_1 = \mathcal{R}_6 \mathcal{R}_1 \mathcal{B}_3 \mathcal{B}_2 \mathcal{B}_1 1 \mathcal{R}_4 \mathcal{R}_2 \mathcal{L}_1 2 \mathcal{R}_3 \mathcal{L}_2 3 \mathcal{B}_7 \mathcal{B}_5 \mathcal{B}_4 \mathcal{A}_3 \mathcal{A}_2 4 \mathcal{R}_5 \mathcal{L}_4 5 \mathcal{B}_6 \mathcal{A}_5 \mathcal{A}_4 \mathcal{A}_1 6 \mathcal{R}_7 \mathcal{L}_6 \mathcal{L}_5 7 \mathcal{L}_7 \mathcal{L}_3 \mathcal{A}_7 \mathcal{A}_6$ .

Let *interval*  $j$  be the sub-sequence between room label  $j$  and room label  $j + 1$ , denoted as  $I(j)$ , and  $|I(j)|$  is the number of positional symbols in  $I(j)$ . Let  $I(0)$  be the sub-sequence before room 1, i.e.  $R(\text{left} - \text{border})B(\text{top} - \text{border})$  and  $I(n)$  be the sub-sequence after room  $n$ , i.e.  $R(\text{right} - \text{border})B(\text{bottom} - \text{border})$ .

The H-sequence is closely related to the Q-sequence which was introduced in [9]: Deleting  $\mathcal{L}(x)$  and  $\mathcal{A}(x)$  from an H-sequence leads to a Q-sequence that corresponds to the same floorplan. Since it is known that the Q-sequence has a one-to-one correspondence to the floorplan, we might say that H-sequence conveys redundant information. However, it will be shown in the following that this redundancy provides us with a merit for heuristic search. The key to this merit is the following fact.

**Theorem 1** *An H-sequence has the following properties.*

1. (*Seg-adjacency*) If  $i$  is a suffix of  $\mathcal{R}$  ( $\mathcal{B}$ ,  $\mathcal{L}$ ,  $\mathcal{A}$ ) in  $I(k)$ , room  $i$  is adjacent to  $s_k$  and at the right (below, left, above) of  $s_k$ .
2. (*Inner-seg-adjacency*) Two rooms  $i$  and  $j$  are channel-adjacent if and only if  $i$  and  $j$  are suffixes of  $\mathcal{R}$ 's ( $\mathcal{L}$ 's) in  $I(k)$  ( $k \neq 0, n$ ), and they are seg-adjacent to and at the right(left) of the vertical seg  $s_k$ . For room label  $i$  and  $j$ , if one is a suffix of an  $\mathcal{R}$  in  $I(k)$ , and the other a suffix of a  $\mathcal{L}$ , then the former is seg-adjacent to  $s_k$  at the right and the later at the left. (Other cases are omitted.)
3. (*Boundary-adjacency*) A room  $i$  adjacent to the left (right, top, bottom) border of the chip (boundary) if and only if  $i$  is a suffix of  $\mathcal{R}$  ( $\mathcal{L}$ ,  $\mathcal{B}$ ,  $\mathcal{A}$ ) in  $I(0)$  or  $I(n)$ .

These properties are all trivial by definition.

## B. Decoding of H-sequence

Only for optimization of routing-cost, we can calculate the routing-cost of the floorplan without decoding the H-sequence to a corresponding floorplan. However, when optimizing routing-cost and area simultaneously, we will decode H-sequence by the decoding procedure for Q-sequence proposed in [9, 4], while the Q-sequence can be obtained from a H-sequence by deleting all  $\mathcal{L}$  and  $\mathcal{A}$ . For example, after deleting  $\mathcal{L}$  and  $\mathcal{A}$  in H-sequence  $H_1$ , we get a Q-sequence,  $Q_1$ :

$$\mathcal{R}_6 \mathcal{R}_1 \mathcal{B}_3 \mathcal{B}_2 \mathcal{B}_1 1 \mathcal{R}_4 \mathcal{R}_2 2 \mathcal{R}_3 3 \mathcal{B}_7 \mathcal{B}_5 \mathcal{B}_4 4 \mathcal{R}_5 5 \mathcal{B}_6 6 \mathcal{R}_7 7$$

## IV. PERTURBATIONS OF PLACEMENT

In order to improve routing-cost or/and chip area, following four perturbations are utilized to perturb a placement.

- (1) **rotation**: Choose one module and rotate it.
- (2) **flip**: Choose one module and flip it horizontally or vertically.
- (3) **swap**: Choose a pair of rooms and exchange the modules inside.
- (4) **moving positional symbols**: With a pair of parenthesis trees of corresponding Q-sequence[4], we can consistently move  $\mathcal{R}$  and  $\mathcal{B}$  positional symbols as well as  $\mathcal{A}$  and  $\mathcal{L}$  positional symbols, based on the relations between positional symbols. Moving positional symbols always leads to a new feasible H-sequence. Detailed description of the relations between  $\mathcal{R}$ ,  $\mathcal{B}$ ,  $\mathcal{A}$ , and  $\mathcal{L}$  are omitted here.

All operations have impacts on the routing-cost calculation. To speed up the iterations, we update incrementally routing-cost of a placement. After rotating or flipping module  $i$ , we need to re-calculate the routing-cost of module  $i$ . After swapping two module  $i$  and module  $j$ , we need to re-calculate the routing-cost of module  $i$  and module  $j$ . For moving positional symbols, we can know which modules' routing-costs should be re-calculated.

It takes  $O(n^2)$  time to re-calculate the routing-cost of the placement at the worst case.

H-sequence keeps the same size of solution space, solution diameter as that of Q-sequence[4].

## V. EXPERIMENTAL RESULTS

Using H-sequence as the floorplan representation, we implemented a placement algorithm based on the simulated annealing in C++ on SPARC 60 workstation. Five MCNC BBL benchmarks are utilized.

Our experiments consist of three parts: *routing-cost optimization*, *restricted routing-cost optimization*, and *simultaneous area and routing-cost optimization*, where the area is that of the minimum bounding box enclosing all modules.

The results of experiments are displayed in Tables I, II, III, where **RC** is routing-cost of the placement, **NOPM** is the number of pairs of modules with the same nets' pins, **NMPC** is the number of matched-pairs facing to the same channel (Case I defined in Section II), and **CRMP** is the *completion ratio* of matched-pairs, i.e.  $\text{NPCB}/\text{NOPR}$ . The units of area and runtime are  $\text{mm}^2$  and *seconds*, respectively.

TABLE I  
RESULTS OF EXPERIMENTS FOR OPTIMIZING ONLY ROUTING-COST

Circuit	RC	Runtime	NMPC	CRMP	NOPM
apte	146	0.3	20	55.56%	36
xerox	313	2.5	25	55.56%	45
hp	117	2.9	21	51.22%	41
ami33	2236	32	214	40.53%	528
ami49	1201	41	137	31.49%	435

**CRMP** in Table I gives out the upper bound of pairs of matched-pairs that can be channel adjacent, which indicates indirectly routability and design quality of the circuit.

As for the optimization only of the channel adjacency, there is an apparent tendency that many modules get together around a few channels as mentioned in Section 1. In order to avoid such impractical solutions, we limit the number of rooms around a seg, which is very easy to implement, since the number of rooms around a seg  $j$  is  $|I(j)|$ , where  $0 \leq j \leq n$ . In experiments, we let the number be less than 8.

From the results on five benchmarks shown in Table II, it was observed that we can get more practical placement while keeping the approximate routing-cost. Since we only optimize routing-cost in Table II and Table I, we need not decode H-sequence to corresponding floorplan, which make the algorithm very fast.

TABLE II  
RESULTS OF RESTRICTING THE NUMBER OF ROOMS AROUND ONE SEG

Circuit	RC	Runtime	NMPC	CRMP
apte	239	0.4	17	47.22%
xerox	334	3.1	25	55.56%
hp	165	3.2	17	41.46%
ami33	2560	40	192	36.36%
ami49	1050	46	125	28.74%

TABLE III  
RESULTS OF OPTIMIZING BOTH ROUTING-COST AND AREA

Circuit	RC	Runtime	Area	NMPC	CRMP
apte	146	0.7	46.92	17	47.22%
xerox	392	5.6	20.38	23	51.15%
hp	224	5.7	10.32	16	39.02%
ami33	2931	59	1.28	119	22.54%
ami49	1266	72	38.65	104	23.91%

The results of experiments for optimizing routing-cost and area in Table III show we can get satisfied area while minimizing routing-cost. Although both shortest total wire length and minimal routing-cost of the placement aim to get a solution easy to route, compared with traditional algorithms that calculate wire length and area simultaneously, our algorithm takes much less time to obtain the optimal results.

## VI. CONCLUSION

We have presented a new floorplan presentation, H-sequence, by which it takes only a constant time to check whether two rooms (modules) are channel-adjacent. Meanwhile, we define a routing-cost to model the routability of a placement. Based on H-sequence and routing-cost, a simulated annealing based placement algorithm can quickly find easy-to-route solutions.

## REFERENCES

- [1] 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, No. 12, pp. 1518–1524, 1996.
- [2] S. Nakatake, H. Murata, K. Fujiyoshi, and Y. Kajitani, "Module packing based on the BSG-Structure and IC layout applications," IEEE Trans. on CAD, Vol. 17, No. 6, pp. 519–530, 1998.
- [3] P. N. Guo, C. K. Cheng, and T. Yoshimura, "An O-Tree representation of non-slicing floorplan and its applications," Proc. of 36th DAC, pp. 286–291, Jun., 1999.
- [4] C. Zhuang, K. Sakanushi, L. Jin, and Y. Kajitani, "An Enhanced Q-Sequence Augmented with Empty-Room-Insertion and Parenthesis Trees," Proc. DATE2002, pp. 61–68, Mar., 2002.
- [5] K. Kozminski and E. Kinnen, "An Algorithm for Finding A rectangular dual of a planar graph for use in area planning for VLSI integrated circuits," Proc. 21st DAC, pp. 655–656, 1984.
- [6] Y-T. Lai and M. Leinwand, "A theory of rectangular dual graphs," Algorithmica, Vol. 5, pp. 467–483, 1990.
- [7] F. Y. Young, D. F. Wong, and H. Young, "Slicing floorplans with boundary constraints," IEEE Trans. on CAD, Vol. 18, No. 9, pp. 1385–1389, Sep., 1999.
- [8] M. Abe, "Covering the square by squares without overlapping," Journal of Japan Mathematical Physics, Vol.4, No.4, pp.359-366, 1930 (in Japanese).
- [9] K. Sakanushi and Y. Kajitani, "The Quarter-State Sequence (Q-Sequence) to represent the floorplan and applications to layout optimization," Proc. of Asia Pacific Conf. on Circuits and Systems 2000, pp. 829–832, Dec., 2000.
- [10] T. Simizu, "Plane dividing T-configurations with consequent numbering and T-symbolism for orthogonal case," Society for Science on Form, Forma, Vol. 5, No. 2, pp. 173–178, 1990.