

Structured Placement with Topological Regularity Evaluation

Shigetoshi NAKATAKE

Department of Information and Media Sciences, University of Kitakyushu
1-1 Hibikino, Wakamatsu, Kitakyushu, Fukuoka 808-0135, Japan
Email: nakatake@env.kitakyu-u.ac.jp

Abstract—This paper introduces a new concept of floorplanning and block placement, called structured placement. Regularity is a key criterion of structured placement so that placement can make progress beyond constraint-driven approaches. This paper formulates the topological regularity that is extractable from a sequence-pair. Regular structures like arrays and rows are defined on a single-sequence that is a kind of standard representation of a sequence-pair. We extract regular structures from a single-sequence in $O(n)$, and then evaluate the structures by quantifying the regularity as an objective function. Besides, we propose a new simulated annealing (SA) framework, called dual SA, where we convey a constructive feature to an SA framework, so that it attains a placement balancing the size of regular structures against the area efficiency. In experiments, we apply our structured placement to analog block designs, and reveal the definite advantage that our placements contain many regular structures such as rows and arrays without increasing the chip area and the wire length, compared to the existing placement.

I. INTRODUCTION

Physical designs of recent analog mixed signal LSIs are facing difficulties due to increasing complexities and request for signal integrity and substrate noise issues. Among them, floorplanning and placement are key technologies in their design flows. In floorplanning, many studies have been reported so far, but especially in recent years, methods based on rectangle packing are widely used such as BSG, Sequence-Pair, O-tree, B*-tree, and TCG-S [2], [3], [4], [5], [7], [12], [14]. Each of them introduces a unique topological system to represent a placement, which is applicable to a stochastic search, for example, simulated annealing. In their experiments, better results along with highly compacted placements have been reported.

However, the consideration of *regularity* lacks in their results, while human designers usually put regular structures into rows and arrays. Actually, such regularity is crucial for analog block placement, since it is believed that this property serves high routability and suppression of variation on performances. In analog placement, the most useful approach is *constraint-driven* [10], [11]. It is possible that this approach yields a good placement to satisfy the design specification if an appropriate constraint is given. However, because it is requested to generate the constraint from both the circuit performance and the layout area efficiency, it is necessary to do these trade-offs in the balance while designing the layout. If the constraint is too severe, the layout area efficiency might be deteriorated. Oppositely, if the constraint is insufficient, the

circuit performance might be degraded. Since no efficient generation of such an appropriate constraint has been established yet, the generation process of the constraint is still manual and time-consuming.

In this paper, we pay attention to the influence of the regularity of placement on the circuit performance, and propose a technique for improving the circuit performance by less constraint.

We formulate *topological regularity* that is extractable from a sequence-pair, and evaluate the regularity during optimization of the placement. In a sense, this concept is *constraint-less*. We refer this concept as *structured placement*, and this is the first paper based on this concept. Also, this paper contributes to extraction and evaluation of topological regular structures of arrays and rows. The regular structures are defined on a single-sequence, which is a kind of standard representation of a sequence-pair. A single-sequence does not need any block label but represents just a topological structure of a placement. First, we introduce an algorithm to extract regular structures from a single-sequence in linear time with respect to the number of blocks.

Next, we evaluate the structures by quantifying the regularity as an objective function. There is no limitation in usage of our extraction and evaluation in ordinary simulated annealing, because the regular structure evaluation can be incorporated into typical objectives such as the area and the wire-length. Thus, our optimization is based on not a constructive approach but a stochastic searching. However, we notice that, in some cases, a constructive approach is successful to explicitly control the structure of the placement. Accordingly, we attempt to take into consideration a constructive feature in a simulated annealing framework by observing our objectives.

Our objectives consists of two properties in the contrast to each other. One is for the topology quality, the other is for quality with respect to physical dimension. Aiming to optimize both properties efficiently, we propose a particular framework of simulated annealing called *dual simulated annealing*. It optimizes topological and physical properties separately and alternately at a step of each temperature, so that its searching converges on a solution of the high regularity as well as of the high compactness.

In experiments, we applied our method to industrial examples for analog block designs. It attained the placements

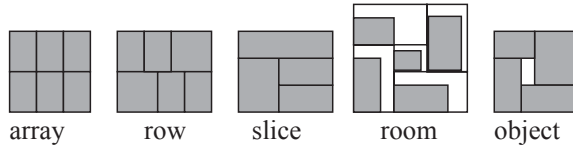


Fig. 1. Structures of placement

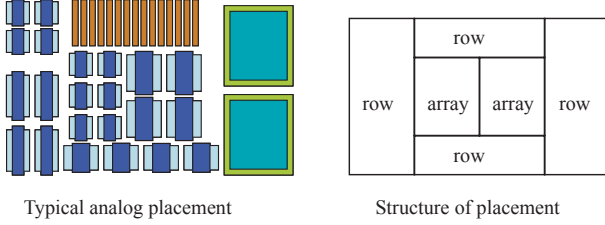


Fig. 2. An image of analog placement and its structure

with a good trade-off in the balance to the regularity and the area efficiency that contained as many rows and arrays as possible without increasing the chip area, compared to those by the existing method. We notice it is likely to be able to use it by combining with the existing constraint-driven approach because there are constraints that must be always satisfied.

The rest of this paper is organized as follows. Section II describes structures of a placement, sequence-pair and single-sequence. Section III and Section IV introduce how to extract regular structures from a single-sequence and how to evaluate the structures, respectively. Section V shows the experimental results. Section VII concludes contributions and future works.

II. PRELIMINARY

A. Regular Structure in Floorplan

Generally speaking, structures of placement can be classified into *array*, *row*, *slice*, *room-base*, and *object-base*. An image of each structure is shown in Fig. 1. Array- and row-structure are often used in gate array and standard cell placement. Slice-structure is seen in [1], called slicing floorplan. Q-sequence [9] and Corner-Block-List [6] stand for room-base-structure that is rectangle dissection of a chip. Rectangle packing based placement such as Sequence-Pair [2] and B*-tree [5] has object-base-structure.

Among these structures, strong or weak relation with respect to the regularity is as $array > row > slice > room > object$. ($A > B$ means A has a stronger regularity than B .) On the contrary, with respect to the flexibility of the area efficiency, the relation is as $array < row < slice < room < object$. Regularity often contributes to suppression of variation of characteristics of same devices, while flexibility does to yield smaller chip area by handling blocks of arbitrary shapes. In a typical placement, strong regularity appears in the local. An image of analog placement and its structure are shown in Fig. 2.

B. Sequence-Pair

Sequence-Pair is introduced in [2]. It is excellent that it defines horizontal and vertical relations independently of

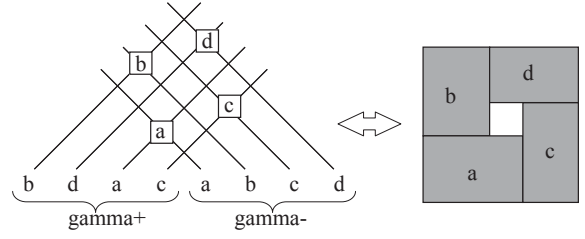


Fig. 3. A sequence-pair representation

each other. In the following, we give a brief introduction of Sequence-Pair.

Given a set of blocks, $\mathcal{B} = \{b_1, b_2, \dots, b_n\}$, a sequence-pair represents topological relations among blocks over a block placement. A sequence-pair is an ordered pair of Γ_+ and Γ_- , where each of Γ_+ and Γ_- is a permutation of names of the given n blocks. For example, $(\Gamma_+, \Gamma_-) = (a, b, c, d; b, d, a, c)$ is a sequence-pair of block set $\{a, b, c, d\}$. If block x is the i -th in Γ_+ , we denote $\Gamma_+(i) = x$, as well as $\Gamma_+^{-1}(x) = i$. Similar notation is also used for Γ_- .

For every block pair (a, b) , a is the left of b (equivalently, b is the right of a) if $\Gamma_+^{-1}(a) < \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a) < \Gamma_-^{-1}(b)$. Analogously, a is below b (equivalently, is above a) if $\Gamma_+^{-1}(a) > \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a) < \Gamma_-^{-1}(b)$.

An example of a sequence-pair and its corresponding placement are shown in Fig. 3.

C. Single-Sequence

In this paper, we focus on a topology of a block placement. The topology can be represented without specifying block's name by using a single-sequence [8], [13]. A single-sequence is denoted by $\mathcal{S} = (s_1, s_2, \dots, s_n) \mid s_k \in \{1, 2, \dots, n\}$. It is defined as $s_k = \Gamma_+^{-1}(\Gamma_-(k))$, that is, \mathcal{S} is the same as Γ_- when each block is renamed as $\Gamma_+ = (1, 2, \dots, n)$. For example, given a sequence-pair, $(\Gamma_+, \Gamma_-) = (b, d, a, c; a, b, c, d)$, the corresponding single-sequence is $\mathcal{S} = (3, 1, 4, 2)$.

III. EXTRACTION OF TOPOLOGICAL REGULAR STRUCTURE

We introduce an extraction algorithm of array- and row-structures. Consider a subsequence X of \mathcal{S} with two or more numbers. Let the minimum and maximum numbers in X be m_X and M_X , respectively. If $M_X - m_X + 1 = |X|$, X is referred as a *rectangular extractable* subsequence. For example, given $\mathcal{S} = (3, 1, 6, 4, 5, 8, 7, 2)$ subsequences $(4, 5)$, $(6, 4, 5)$, $(6, 4, 5, 8, 7)$, and $(3, 1, 6, 4, 5, 8, 7, 2)$ are rectangular extractable.

We introduce *horizontal* and *vertical single rows* as follows; A rectangular extractable X such that $s_{k+1} - s_k = 1$ ($s_{k+1}, s_k \in X$) is a *horizontal single row*. Similarly, X such that $s_{k+1} - s_k = -1$ is a *vertical single row*.

If a rectangular extractable subsequence X is composed of only two or more horizontal single rows that are stacked vertically, X represents the topology of *horizontal multi-row*. Analogously, *vertical multi-row* is defined.

Furthermore, in a multi-row subsequence X , each of rows has the same length, X composes a topology of an *array*.

Theorem 3.1: (Row and Array Extraction) Given a sequence-pair, all multi-rows and arrays with the maximal length that are included in the sequence-pair can be extracted in time complexity $O(n)$, where n is the number of blocks.

Proof: We can convert a sequence-pair to a single-sequence S in time complexity $O(n)$. We prove the theorem by providing a procedure to extract rows and arrays from S followed by an example. First, we describe an extraction of horizontal multi-rows.

- 1) Divide S into subsequences $S_1/S_2/\dots/S_i/\dots/S_l$ such that $s_k - s_{k+1} = 1$ where $s_k, s_{k+1} \in S_i$. For example, given $S = (1, 2, 7, 8, 9, 5, 6, 3, 4, 10)$, the subsequences are $(1, 2)/(7, 8, 9)/(5, 6)/(3, 4)/(10)$.
- 2) For each subsequence S_i , let the minimum and maximum numbers be m_{S_i} and M_{S_i} , respectively. In the example, $\{(m, M)\} = \{(1, 2), (7, 9), (5, 6), (3, 4), (10, 10)\}$.
- 3) For each adjacent pair of subsequences, S_i and S_{i+1} , we say they are *vertical stackable* if $m_{S_i} - M_{S_{i+1}} = 1$. Extract all vertical stackable pairs. The vertical stackable pairs in the example are $((7, 8, 9), (5, 6))$ and $((5, 6), (3, 4))$.
- 4) Concatenate two or more stackable pairs if they have a common subsequence. As the result, the concatenated subsequence corresponds to a horizontal multi-row. In the example, $((7, 8, 9), (5, 6), (3, 4))$ forms a multi-row. Note that if the multi-row has the same length of each single-row, it corresponds to an array.

In step 1), we obtain a set of horizontal single-rows. In step 3), since $m_{S_i} - M_{S_{i+1}} = 1$, $m_{S_i} > M_{S_{i+1}}$. This means $\forall s_u \in S_i > \forall s_v \in S_{i+1}$. Then, all blocks in S_{i+1} are placed on all in S_i .

Let S_i and S_{i+1} be $(a, a+1, a+2, \dots, a+p)$ and $(b, b+1, b+2, \dots, b+q)$, respectively. Because $m_{S_i} = a$ and $M_{S_{i+1}} = b+q$, $m_{S_i} - M_{S_{i+1}} = a - (b+q) = 1$, that is, $a = (b+q) + 1$. It is proved that (S_i, S_{i+1}) is rectangular extractable as follows;

$$\begin{aligned} & \max(M_{S_i}, M_{S_{i+1}}) - \min(m_{S_i}, m_{S_{i+1}}) + 1 \\ &= \max(a+p, b+q) - \min(a, b) + 1 \\ &= (a+p) - b + 1 = ((b+q) + 1) + p - b + 1 \\ &= p + 1 + q + 1 = |S_i| + |S_{i+1}|. \end{aligned}$$

Therefore, S_i and S_{i+1} compose double horizontal single-rows stacked vertically, and their stacked structure is rectangular extractable, that is, a multi-row.

Next, consider a concatenation of a multi-row $X = X_1/X_2/\dots/X_k$ and a subsequence S_i . X and S_i are adjacent in S as shown in $\dots/X_1/X_2/\dots/X_k/S_i/\dots$. Note that, since X is a multi-row, $M_{X_1} \geq m_{X_1} > M_{X_2} \geq m_{X_2} > \dots > M_{X_k} \geq m_{X_k}$.

From the condition in step 3), since $m_{X_k} - M_{S_i} = 1$, $m_{X_k} > M_{S_i}$, that is, $M_X \geq m_X > M_{S_i} \geq m_{S_i}$. Hence, all blocks in S_i are placed on all in X .

More, $M_X - m_X + 1 = |X|$ and $m_{X_k} - M_{S_i} = m_X - M_{S_i} = 1$. By the similar discussion as the above,

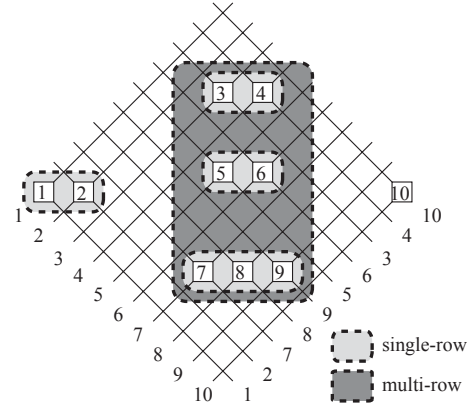


Fig. 4. An example of extraction of row structures

$$\begin{aligned} & \max(M_X, M_{S_i}) - \min(m_X, m_{S_i}) + 1 \\ &= M_X - m_{S_i} + 1 = (|X| + m_X - 1) - m_{S_i} + 1 \\ &= |X| + (M_{S_i} + 1) - m_{S_i} = |X| + |S_i|. \end{aligned}$$

Therefore, X and S_i are stacked vertically, and their stacked structure is rectangular extractable.

Oppositely, assume that a multi-row $X = \{S_i\}$ with the maximal length is given, where S_i is a horizontal single-row. S_{i+1} is stacked on S_i vertically. Then, $M_{S_i} \geq m_{S_i} > M_{S_{i+1}} \geq m_{S_{i+1}}$.

$$\begin{aligned} & \max(M_{S_i}, M_{S_{i+1}}) - \min(m_{S_i}, m_{S_{i+1}}) + 1 = |S_i| + |S_{i+1}|. \\ & M_{S_i} - m_{S_{i+1}} + 1 = (M_{S_i} - m_{S_i} + 1) + (M_{S_{i+1}} - m_{S_{i+1}} + 1). \\ & m_{S_i} - M_{S_{i+1}} = 1. \end{aligned}$$

This is why the above procedure attains the multi-row of the maximal length.

Analogously, vertical multi-rows can be obtained. In an extraction of vertical multi-rows, we divide S into subsequences such that $s_k - s_{k+1} = -1$ in step 1). Besides, in step 3), we extract *horizontal stackable* pairs such that $M_{S_i} - m_{S_{i+1}} = 1$ for S_i and S_{i+1} .

Obviously, each step needs the time complexity that is linear to the number of blocks. Then the theorem is proved. ■

An illustration of the example used in the proof is shown in Fig. 4.

IV. EVALUATION OF REGULARITY

Once we extract multi-rows and arrays as described in the previous section, we evaluate their structures as follows.

A. Topological Structure Value

We introduce a *topological structure value*, denoted by V_{top} , which is measured in terms of sizes and shapes of structures.

Let sets of arrays and multi-rows be A and R , respectively. If a multi-row $r \in R$ has $k \times l$ structure, the aspect is figured out by $\sigma(r) = \min(k, l) / \max(k, l)$. Analogously, we figure out the aspect $\sigma(a)$ of an array $a \in A$.

We define the topological structure value V_{top} as follows;

$$V_{top} = \alpha * \sum_{r \in R} \sigma(r) + \beta * \sum_{a \in A} \sigma(a),$$

where α and β are coefficients to balance a trade-off among structure values.

B. Physical Dimension Cost

We also introduce a *physical dimension cost*, denoted by C_{phy} , which consists of *local compactness* and *local uniformity*. Note that we can calculate the cost before a compaction process of sequence-pair decoding.

In a multi-row r , a block assigned to the slot of the i -th row and the j -th column is denoted by $r_{i,j}$. Also, the width, the height and the area of $r_{i,j}$ are denoted by $w(r_{i,j})$, $h(r_{i,j})$, and $a(r_{i,j})$, respectively.

In the following, we describe the definition of the local compactness and the local uniformity as for a multi-row. These costs for arrays can be analogously defined, but it is omitted here for the space.

1) *Local compactness of a multi-row r* , $C_{cmp}(r)$: If r is composed of horizontal single rows,

$$C_{cmp}(r) = \max_i \sum_j (w(r_{i,j})) * \sum_j \max_j (h(r_{i,j})) - \sum_{i,j} a(r_{i,j}).$$

Otherwise,

$$C_{cmp}(r) = \sum_j \max_i (w(r_{i,j})) * \max_j \sum_i (h(r_{i,j})) - \sum_{i,j} a(r_{i,j}).$$

2) *Local uniformity of a multi-row r* , $C_{uni}(r)$: If r is composed of horizontal single rows,

$$C_{uni}(r) = \sum_i \max_j w(r_{i,j}) - \min_j w(r_{i,j}).$$

Otherwise,

$$C_{uni}(r) = \sum_j \max_i h(r_{i,j}) - \min_i h(r_{i,j}).$$

Finally, we define physical dimension cost C_{phy} as follows;

$$C_{phy} = \alpha' * \sum_{a \in A} (C_{cmp}(a) + C_{uni}(a)) + \beta' * \sum_{r \in R} (C_{cmp}(r) + C_{uni}(r))$$

where α' and β' are coefficients to balance a trade-off among costs.

V. DUAL SIMULATED ANNEALING

There is no limitation in usage of our extraction and evaluation in an ordinary simulated annealing as long as V_{top} and C_{phy} are incorporated into the objective such as area and wire-length. However, sometimes, we may need somewhat constructive features to successfully control structures of a placement. In this section, we convey a constructive feature to a simulated annealing framework.

A. Framework

Our original objectives are V_{top} and C_{phy} , that is, one is for the topology, the other is for the physical dimension. Aiming to optimize both objectives efficiently, we propose *dual simulated annealing*. It optimizes topological and physical objectives separately and alternately at a step of each temperature. The framework is as follows.

- 1: SP := GenerateInitialSP();
- 2: P := GeneratePlacement(SP);
- 3: **for** (Temperature decreasing) **do**

- 4: E := $E_{top}(P)$;
- /* topological optimization */
- 5: **for** (Topological structure searching) **do**
- 6: SP1 := MoveTopologicalStructure(SP);
- 7: P := GeneratePlacement(SP1);
- 8: E1 := $E_{top}(P)$;
- 9: **if** IsAccept(temp, E, E1) **then**
- 10: SP := SP1;
- 11: E := E1;
- 12: **end if**
- 13: **end for**
- 14: P := GeneratePlacement(SP);
- 15: E := $E_{phy}(P)$;
- /* physical optimization */
- 16: **for** (Physical structure searching) **do**
- 17: SP1 := MovePhysicalStructure(SP);
- 18: P := GeneratePlacement(SP1);
- 19: E1 := $E_{phy}(P)$;
- 20: **if** IsAccept(temp, E, E1) **then**
- 21: SP := SP1;
- 22: E := E1;
- 23: KeepBestSoFar(SP, E);
- 24: **end if**
- 25: **end for**
- 26: **end for**

B. Cost Function

In the above framework, we use two cost functions, E_{top} and E_{phy} . Given a placement P along with its sequence-pair, we calculate the primary objective such as the chip area or the wire length. We describe about the primary objective in experiments, but it is denoted by $F(P)$. Then, both functions, E_{top} and E_{phy} , are calculated as follows;

$$E_{top}(P) = \frac{F(P)}{g(V_{top})}, E_{phy}(P) = F(P) * g(C_{phy}).$$

g is a conversion that maps an input value to another value within [1.0, 1.1], and it is defined as;

$$g(x) = 1.0 + 0.1 \exp\left(\frac{-x_m \log(0.5)}{x + \epsilon}\right),$$

where x_m is an average value of $\{x\}$ and ϵ is a small value enough to ignore. The meaning of the function g is to be likely to degrade $F(P)$ by 10% to obtain better topological structure value or less physical dimension cost.

C. Move

In the above framework, the topology is optimized between line 5-13, while the physical dimension is done between line 16-25. We apply different moves to the different optimization, *MoveTopologicalStructure* and *MovePhysicalStructure*.

1) *MoveTopologicalStructure*: We use *HalfExchange* of a sequence-pair to generate another sequence-pair, which is introduced in [2]. This move is a pair-exchange of blocks on either Γ_+ or Γ_- .

2) *MovePhysicalStructure*: We apply *FullExchange* and *HalfExchangeKeepingStructure* to a sequence-pair. The former is the same as in [2]. The latter is designed to keep the topologies of multi-rows and arrays. Since a multi-row and an array are both rectangular extractable, each of structures

corresponds to a subsequence of the sequence-pair. Then, we apply a subsequence-exchange keeping block orders in the subsequences.

VI. EXPERIMENTS

We implemented our structured placement that is composed of the topology regularity extraction, the regularity evaluation, and the optimization by dual simulated annealing.

A. Analog Block Designs

We tested our placement tool for analog block designs. We prepared 13 instances of analog block sets from industries. The number of blocks and that of nets of each data are shown in Table I.

To make our contribution distinct, we also implemented a normal block placement based on sequence-pair. This placement optimizes the primary objective described later by a standard simulated annealing. The same annealing schedule was used for the both placements. And then, we compared to each other with respect to the chip area, the wire length, and the structure coverage. The structure coverage is the ratio to the total number of blocks of the number of blocks composing topological arrays or rows.

First, we set the primary objective as the chip area. The numerical data of the results is shown in Table I. For all instances, the area ratio that is to the area by the normal placement of that by our structured placement is less than 5%, while the structure coverage of the normal placement and that of our structured placement are quite different 7.9% and 78.3% on the average, respectively. The resultant placements of data A, B, and C by both tools are shown in Fig. 5.

Second, we set the primary objective as the product of the chip area and the wire length. The numerical data of the results is shown in Table II. Compared the results by our placement to those by normal placement, the ratio on the product of the chip area and the wire length is 0.978 on the average. Besides, the structure coverages are 9.6% and 73.5% on the average, respectively. The resultant placements of data I and L by both tools are shown in Fig. 6.

It is observed that our placement is successful for the primary objective while the cost function is designed to contain a factor to degrade the primary objective. The reason can be guessed that pair-exchanging of structures in dual SA enable a solution to escape from a local minima. An observation of the similar kind is seen in the cluster-constraint-driven approach. However, notice that our structured placement did not use any cluster constraint. Actually, we are convinced that it is necessary to combine the constraint-driven method with our structured placement to develop a more practical tool.

VII. CONCLUSION

We introduced a new concept, structured placement, which makes use of the regularity of topological structure as a key criterion. The regularity is formulated in terms of topological arrays and rows. We provided the extraction of the regular structures from a single-sequence in $O(n)$, as well as the way

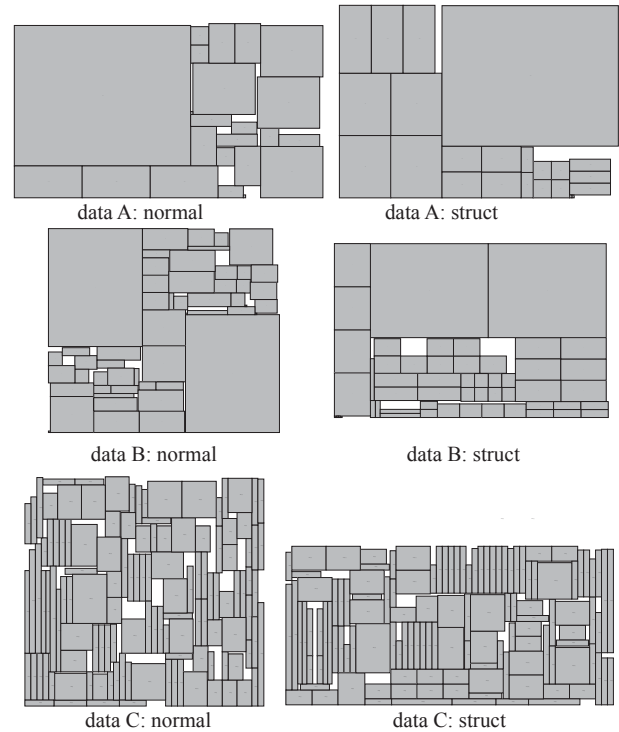


Fig. 5. Resultant placements of data A, B, and C, where the primary objective is the chip area: “normal” and “struct” are normal placement and our structured placement, respectively

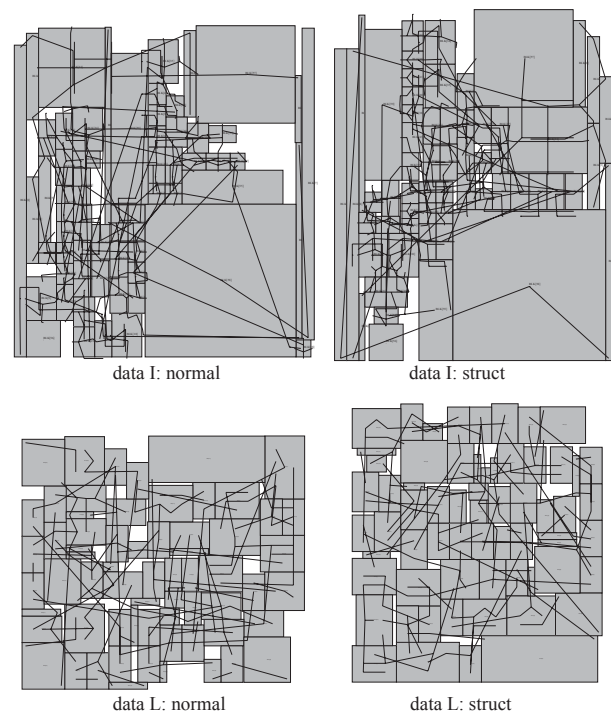


Fig. 6. Resultant placements of data I and L, where the primary objective is the product of the chip area and the wire length

TABLE I

NUMERICAL DATA AND RESULTS OF ANALOG BLOCK DESIGNS, WHERE THE PRIMARY OBJECTIVE IS THE CHIP AREA: "NORMAL" AND "STRUCT" ARE NORMAL PLACEMENT AND OUR STRUCTURED PLACEMENT, RESPECTIVELY. "STR-COVER" IS THE RATIO TO THE TOTAL NUMBER OF BLOCKS OF THE NUMBER OF BLOCKS COMPOSING TOPOLOGICAL ARRAYS OR ROWS. "AREA RATIO" IS THE RATIO TO THE AREA BY "NORMAL" OF THAT BY "STRUCT".

data	# blocks	# nets	normal		struct		area ratio struct/normal
			area (μm^2)	str-cover (%)	area (μm^2)	str-cover (%)	
A	23	44	432,876	9 (%)	432,595	96 (%)	1.00
B	53	90	848,114	0 (%)	857,419	96 (%)	1.01
C	122	91	98,868	5 (%)	93,720	72 (%)	0.95
D	60	46	75,078	17 (%)	75,594	87 (%)	1.01
E	113	80	238,392	4 (%)	235,128	68 (%)	0.99
F	32	22	65,367	19 (%)	68,150	69 (%)	1.04
G	54	49	63,248	15 (%)	64,904	72 (%)	1.03
H	90	58	87,870	9 (%)	88,960	77 (%)	1.01
I	60	36	10,265	3 (%)	10,192	82 (%)	0.99
J	101	78	39,619	5 (%)	40,591	84 (%)	1.02
K	66	29	168,866	12 (%)	170,990	79 (%)	1.01
L	64	49	53,710	0 (%)	54,108	88 (%)	1.01
M	166	105	72,945	5 (%)	71,714	49 (%)	0.98

TABLE II

NUMERICAL RESULTS OF ANALOG BLOCK DESIGNS, WHERE THE PRIMARY OBJECTIVE IS THE PRODUCT OF THE CHIP AREA AND THE WIRE LENGTH: "WIRE-LEN" IS THE WIRE LENGTH. "AREA*WIRE-LEN RATIO" IS THE RATIO TO THE AREA*WIRE-LEN BY "NORMAL" OF THAT BY "STRUCT"

data	normal			struct			area*wire-len ratio struct/normal
	area (μm^2)	wire-len (μm)	str-cover (%)	area (μm^2)	wire-len (μm)	str-cover (%)	
A	444,566	14,488	26 (%)	463,810	15,717	87 (%)	1.13
B	970,759	43,554	11 (%)	973,116	44,693	71 (%)	1.03
C	104,895	9,326	5 (%)	99,231	10,612	64 (%)	1.08
D	81,073	5,794	17 (%)	79,605	6,250	62 (%)	1.06
E	261,332	19,926	7 (%)	296,800	13,985	73 (%)	0.80
F	67,680	2,632	13 (%)	72,380	2,491	91 (%)	1.01
G	72,192	3,750	15 (%)	68,906	3,241	87 (%)	0.82
H	97,280	9,412	9 (%)	89,694	9,249	66 (%)	0.91
I	10,779	2,413	7 (%)	10,554	2,205	78 (%)	0.89
J	41,800	3,695	2 (%)	40,460	3,779	71 (%)	0.99
K	173,870	3,529	3 (%)	173,040	3,620	64 (%)	1.02
L	58,629	5,603	9 (%)	57,002	5,361	83 (%)	0.93
M	77,451	12,997	1 (%)	80,575	13,126	58 (%)	1.05

to evaluate the regular structures. Also, we proposed a new SA framework, called dual SA, where we convey a constructive feature to an SA framework. In experiments in analog block designs, the results by our structured placement are arranged in an orderly manner, that is, it is composed of many regular structures, without increasing the chip area and the wire length, compared to the existing placement.

We are convinced that there is no limitation in usage of our structured placement in general placement and floorplanning framework. In future works, we will apply the structured placement to analog floorplanning along with further practical extensions.

REFERENCES

- [1] D. F. Wong and C. L. Liu, *A new algorithm for floorplan design*, Proc. 23rd DAC, pp.101–107, 1986.
- [2] H. Murata, S. Nakatake, K. Fujiyoshi, and Y. Kajitani, *VLSI module placement based on rectangle-packing by Sequence-Pair*, IEEE Trans. on CAD, Vol.15, No.12, pp.1518–1524, 1996.
- [3] 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, No.6, pp.519–530, 1998.
- [4] P. N. Guo, C. K. Cheng, and T. Yoshimura, *An O-tree representation of non-slicing floorplan and its applications*, Proc. 36th DAC, pp.268–273, 1999.
- [5] Y-C. Chang, Y-W. Chang, and G-M. Wu, *B*-trees: A new representation for non-slicing floorplan*, Proc. 37th DAC, pp.458–463, 2000.
- [6] X. Hong, S. Dong, Y. Ma, Y. Cai, C. K. Cheng, and J. Gu, *Corner Block List: An efficient topological representation of non-slicing floorplan*, Proc. ICCAD 2000, pp.8–12, 2000.
- [7] J. Lin and Y. Young, *TCG-S: Orthogonal coupling of P*-admissible representations for general floorplans*, Proc. 39th DAC, pp.842–847, 2002.
- [8] C. Kodama and K. Fujiyoshi, *Selected Sequence-Pair: An efficient decodable packing representation in linear time using Sequence-Pair*, Proc. ASP-DAC 2003, pp.331–337, 2003.
- [9] K. Sakanushi, Y. Kajitani, and Dinesh Mehta, *The quarter-state sequence floorplan representation*, IEEE Trans. on CAS-I, Vol.50, No.3, pp.376–386, 2003.
- [10] T. Nojima, X. Zhu, Y. Takashima, S. Nakatake, and Y. Kajitani, *Multi-level placement with circuit schema based clustering*, Proc. ASP-DAC 2004, pp.406–411, 2004.
- [11] T. Nojima, Y. Takashima, S. Nakatake, and Y. Kajitani, *A device-level placement with multi-directional convex clustering*, Proc. GLSVLSI 2004, pp.196–201, 2004.
- [12] J. M. Lin and Y. W. Chang, *TCG-S: Orthogonal coupling of P*-admissible representations for general floorplans*, IEEE Trans. on CAD, Vol.24, No.6, pp.968–980, 2004.
- [13] X. Zhang and Y. Kajitani, *Theory of T-junction floorplans in terms of single-sequence*, Proc. ISCAS 2004, pp.341–344, 2004.
- [14] J.-M. Lin and Y.-W. Chang, *A transitive closure graph based representation for general floorplans*, IEEE Trans. on VLSI Systems, Vol.13, No.2, pp.288–292, 2005.