IPSJ Transactions on System LSI Design Methodology Vol. 2 222–238 (Aug. 2009)

Regular Paper

Structured Placement with Topological Regularity Evaluation

Qing Dong†1 and Shigetoshi Nakatake†1

This paper introduces a new concept of regularity-oriented floorplanning and block placement—structured placement, it takes the regularity as a criterion of placement so as to improve the performance. We provide the methods to extract regular structures from a placement representation in linear time, and manage to evaluate these structures by quantifying the regularity as an objective function. We also construct a particular simulated annealing framework, which optimizes placement topology and physical dimension separately and alternately so that it attains a solution balancing the trade-off between regularity and area efficiency. Furthermore, we introduce the symmetry-oriented structured placement to produce symmetrical placement. Experiments show that the resultant placements achieve regularity without increased chip area and wire length, compared to those by existing methods.

1. Introduction

As the system on a chip (SoC) technology progresses in recent years, both digital and analog functional units are implemented on the same chip. Thermal, parasitics, signal integrity and substrate issues are all taken into account, so physical design tools should offer both digital and analog capabilities to the mixed-signal integrated circuits. As an important process in the design flow, the placement tool must not only provide a good compact layout result, but also cover analog specific features, such as *regularity*. Regularity is crucial for analog placement since it contributes to high routability and suppression of variation on performance. An analog placement with less regularity might introduce more bends and vias, consequently increase the interconnect parasitics, sometimes even risk the circuit a disastrous performance. **Figure 1** shows how the regularity affects the routing channels. The routing channels in A, B and C are extracted

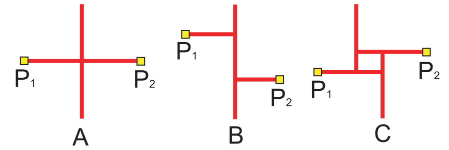


Fig. 1 Routing channels with different regularity.

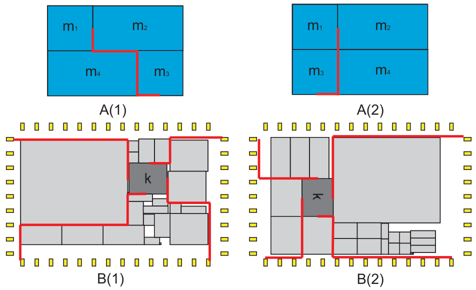


Fig. 2 Routing bends reduction by the placement with better regularity. Placement $A(2)$ and $B(2)$ have better regularity than placement $A(1)$ and $B(1)$, thereby are more preferable.

from the same benchmark with different regularities. Since A has the best regularity, therefor for a net connecting pin P_1 and P_2, the number of routing bends in A is the smallest.

Figure 2 explains why a placement with good regularity is preferable. Two placements for the same problem A are shown in Fig. 2, i.e., $A(1)$ and $A(2)$, and $A(2)$ has better regularity than $A(1)$. Denote the top, bottom, left and right edge of a block m as, m_T, m_B, m_L and m_R respectively. Given a two-pin net connecting pin S in block m_1 to pin T in m_3, there are 16 ways to assign S and T into different edges of corresponding blocks. For $A(2)$ there are 3 assignments that can introduce non-bend routing, i.e., assign S into m_1_B and T into m_3_T; assign S into m_1_L and T into m_3_L; assign S into m_1_R and T into m_3_R. While for $A(1)$ there is only one assignment, i.e., assign S into m_1_B and T into m_3_T, that can introduce non-bend routing. Furthermore, for those assignments which can not avoid bends, $A(2)$ still exceeds $A(1)$. Red bold segments demonstrate the routing when assigning S into m_1_R and T into m_3_L, and the bends in $A(1)$ and $A(2)$ are 3 and 1 respectively. The problem B in Fig. 2 is a more practical case

†1 The University of Kitakyushu

extracted from an industrial circuit. We can see that $B(2)$ has better regularity than $B(1)$. To explain why $B(2)$ is more preferable, without loss of generality, we select a block k in the problem (colored in dark grey), and assume that k has 4 pins connected to 4 pads located at 4 corners of the chip respectively by 4 nets. The pins of k are assigned so that the total routing bends of the nets in $B(1)$ are minimized (see Fig. 2, there are 8 bends in $B(1)$). However, we can see that, even in such an assignment, the total number of bends of the 4 nets in $B(2)$ are still less than that of $B(1)$ (total 5 bends in $B(2)$).

Many techniques for block placement have been studied in recent years, methods based on rectangle packing such as BSG, Sequence-pair, O-tree, B*-tree, and TCG-S [3),5),8)–10),13),15)] are widely used. These methods devote into generating placement with less area and wire length, while none of them concerns the regularity. Even the constraint-driven approaches [11),16)] for analog placement just cope with the specified devices to suppress mismatch, they do not care the regularity neither.

In this paper, we introduce a new concept of placement – *structured placement*, it takes the regularity as a criterion of placement. With the proposed method, the traditional goals of placement such as less area and wire length are preserved as usual, meanwhile a placement can pursue the regularity automatically. Our methods can easily cooperate with constraint-driven approaches to complement the analog placement, constraints take care of specified devices, our approach takes care of the rest part of a placement. In experiments, we applied our structured placement to industrial instances for analog block designs. The experiments show that the resultant placements contain as many regular structures as possible without increased chip area and wire length. Symmetry-oriented structured placement demonstrates its excellence by producing symmetrical results without loss of concern about regularity. Our contributions are summarized as follows.

- We introduce the concept of *structured placement*. By quantifying and incorporating the topological regularity into evaluation, we make a placement achieve regularity.
- We propose the methods to extract regular structures from a placement and evaluate their regularities in linear time with respect to the number of blocks.
- We propose the *dual simulated annealing* procedure to optimize the placement's topological and physical properties separately and alternately, so the searching converges at a solution with high regularity as well as high compactness.
- We also propose a symmetry-oriented optimization procedure to generate symmetry-oriented structured placement. We formulate the topological complete symmetry structure, and design efficient move operations to maintain the placement topological symmetry during the optimization.

The rest of this paper is organized as follows. Section 2 defines the problem. Section 3 describes the structures of a placement and the representations used in this paper. Section 4 introduces the structured placement including the extraction and evaluation of topological structures, and the optimization procedure. Section 5 introduces the symmetry-oriented structured placement. Section 6 shows the experimental results. Section 7 concludes contribution and future works.

2. Problem Formulation

Given a set of n rectangular blocks $B = b_1, \ldots, b_n$, a set of m nets $N = N_1, \ldots, N_m$ between the n blocks. w_i and h_i are the width and height of each block respectively, where $i = 1, \ldots, n$. The bottom-left corner of each block is denoted by (x_i, y_i). We assume that the bottom-left corner and the top-right corner of chip locate at $(0, 0)$ and $(max\{x_i + w_i\}, max\{y_i + h_i\})$ respectively. A placement is an assignment of (x_i, y_i) for each block such that no two blocks overlap and all blocks are placed in the chip area. The objective of our work is to generate a placement P with good regularity meanwhile minimize the chip area and total wire length. The details of regularity and the cost function will be presented in Section 4.

3. Topological Structure of Placement

3.1 Regular Structure of Placement

Generally speaking, structures of placement can be classified into *array*, *row*, *symmetry*, and *random*, as shown in **Fig. 3**. Array- and row-structure are often used in gate array and standard cell placement. Symmetry-structure is often requested in analog layout for device matching. Rectangle packing based place-

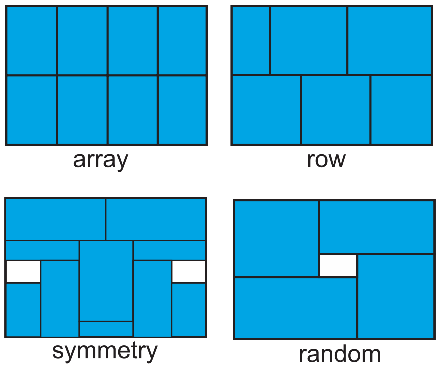


Fig. 3 Structures of placement.

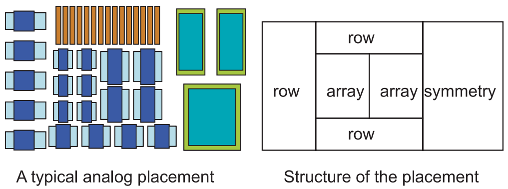


Fig. 4 An image of analog placement and its structure.

ment such as Sequence-Pair [13] and B*-tree [3] has random-structure. Among these structures, strong or weak relation with respect to the regularity is as *array* $>$ *row* $>$ *symmetry* $>$ *random*. ($A > B$ 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* $<$ *symmetry* $<$ *random*. 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 is shown in **Fig. 4**.

3.2 Sequence-Pair & Single-Sequence

A sequence-pair [12] is an ordered pair of Γ_+ and Γ_- to represent a placement. Each of Γ_+ and Γ_- is a permutation of names of given n blocks. 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 left to b (equivalently, b is right to a) if $\Gamma_+^{-1}(a) < \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a) < \Gamma_-^{-1}(b)$. Analogously, a is below b (equivalently, b is above a) if $\Gamma_+^{-1}(a) > \Gamma_+^{-1}(b)$ and $\Gamma_-^{-1}(a) < \Gamma_-^{-1}(b)$.

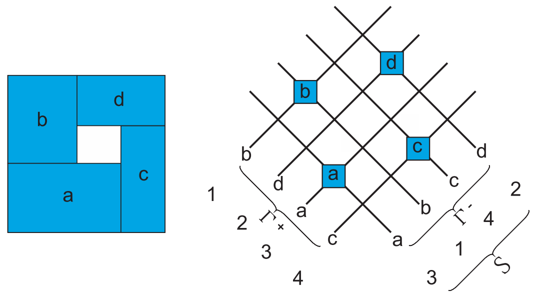


Fig. 5 A placement with its corresponding sequence-pair and single-sequence.

The single-sequence [7),17] can represent a placement's topology without specifying block's name. It is defined as $\mathcal{S}(k) = \Gamma_+^{-1}(\Gamma_-(k))$, that is, $\mathcal{S}$ is the same as Γ_- when each block is renamed as $\Gamma_+ = (1, 2, \ldots, n)$. For example, given a sequence-pair, $(\Gamma_+, \Gamma_-) = (b, d, a, c; a, b, c, d)$, the corresponding single-sequence is $\mathcal{S} = (3, 1, 4, 2)$.

An example of a placement with its corresponding sequence-pair and single-sequence are shown in **Fig. 5**.

4. Structured Placement

4.1 Extraction of Regular Structures

Firstly, we introduce the extraction algorithm of array- and row-structure from a single-sequence $\mathcal{S}$. Consider a subsequence X of the $\mathcal{S}$ with two or more numbers. Let the minimum number and the maximum number 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 $\mathcal{S} = (1, 2, 7, 8, 9, 5, 6, 3, 4, 10)$, subsequences $(1, 2), (7, 8), (8, 9), (5, 6)$, and $(3, 4)$ are rectangular extractable, as well as $(7, 8, 9)$,$(7, 8, 9, 5, 6)$, $(5, 6, 3, 4)$, $(7, 8, 9, 5, 6, 3, 4)$ and $(1, 2, 7, 8, 9, 5, 6, 3, 4)$.

Horizontal and *vertical single rows* are defined as follows; A rectangular extractable X such that $x_{k+1} - x_k = 1$ $(1 \leqslant k \leqslant |X|)$ is a *horizontal single row*. Similarly, X such that $x_{k+1} - x_k = -1$ $(1 \leqslant k \leqslant |X|)$ is a *vertical single row*. The row length is defined by $|X|$.

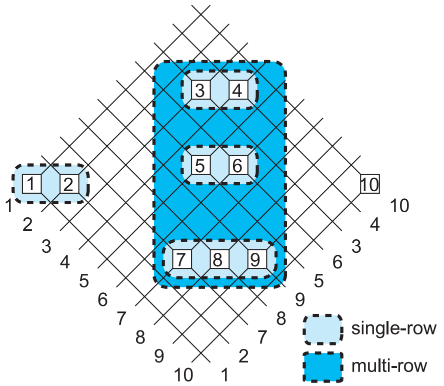


Fig. 6 An example of the extraction of row structures.

If a rectangular extractable subsequence X is composed of two or more horizontal single rows that are stacked vertically, X represents the topology of *horizontal multi-row. vertical multi-row* is defined analogously. Furthermore, in a multi-row subsequence X, if each row has the same length, X composes a topology of an *array.* The length of a multi-row or an array is defined as the number of single rows.

The extraction of horizontal multi-rows is described as follows. The illustration of an example is shown in **Fig. 6**.

(1) Convert a given sequence-pair into corresponding single sequence $\mathcal{S}$.

(2) Divide $\mathcal{S}$ into subsequences $S_1/S_2/\cdots/S_i/\cdots/S_l$ such that $s_{k+1}-s_k=1$ where $s_k, s_{k+1} \in S_i$. For example, given $\mathcal{S}=(1,2,7,8,9,5,6,3,4,10)$, the subsequences are $(1,2)/(7,8,9)/(5,6)/(3,4)/(10)$.

(3) For each subsequence S_i, let the minimum number and the maximum number be m_{S_i} and M_{S_i}, respectively. In the example, $\{(m,M)\} = \{(1,2),(7,9),(5,6),(3,4),(10,10)\}$.

(4) 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 $((5,6),(3,4))$.

(5) 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 each single-row has the same length, the multi-row corresponds to an array.

Theorem 4.1 (Row and Array Extraction) Given a sequence-pair, all the 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 amount of blocks.

proof: First, we prove that the procedure described above can extract all multi-rows and arrays with the maximal length from the sequence-pair.

In step (2), a set of horizontal single-rows is obtained. In step (4), 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 above all in S_i.

Let S_i and S_{i+1} be $(a, a+1, a+2, \ldots, a+p)$ and $(b, b+1, b+2, \ldots, b+q)$, 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 two horizontal single-rows stacked vertically, and their stacked structure is rectangular extractable, that is, a multi-row.

Then, consider a concatenation of a multi-row $X = X_1/X_2/\cdots/X_k$ and a subsequence S_i. S_i, X as well as $X_j (j=1,\ldots,k)$ are the subsequences of S. X and S_i are adjacent in $\mathcal{S}$ as $\cdots/X_1/X_2/\cdots/X_k/S_i/\cdots$. Note that, since X is a multi-row, $M_{X_1} \geq m_{X_1} > M_{X_2} \geq m_{X_2} > \ldots > M_{X_k} \geq m_{X_k}$.

From the condition in step (4), 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 the blocks in S_i are placed above all the blocks in X.

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

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

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

Oppositely, for a multi-row $X = S_p/S_{p+1}/\ldots/S_{p+q-1}$, assume that q is the maximal extractable length starting from S_p, i.e. q is the row number of X, and assume that each S_i $(i = p, \ldots, p+q-1)$ is a horizontal single-row, then S_{i+1} $(i = p, \ldots, p+q-2)$ is stacked on S_i vertically, and $M_{S_i} \geq m_{S_i} > M_{S_{i+1}} \geq m_{S_{i+1}}$, so

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

So S_i and S_{i+1} is recognized as *vertical stackable* pair in the step (4). Step (5) concatenates all the subsequences in X into one sequence corresponding to a multi-row, with the largest number of stacked single rows. Then X is extracted with the maximal length.

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

Analogously, vertical multi-rows can be obtained. In the extraction of vertical multi-rows, $\mathcal{S}$ is divided into subsequences such that $s_{k+1} - s_k = -1$ in step (2). Besides, in step (4), we can also extract *horizontal stackable* pairs such that $M_{S_i} - m_{S_{i+1}} = -1$ for S_i and S_{i+1}.

Then we explain the linear time complexity.

A sequence-pair can be converted into a single-sequence in linear time to the amount of blocks. So step (1) has the time complexity of $O(n)$.

Step (2) also needs an $O(n)$ time complexity. After step (2), the $\mathcal{S}$ will be divided into subsequences, the amount of subsequences is denoted by n_{sub}, note that $n_{sub} \leqslant n$. The amount of blocks in each subsequence S_i is denoted by n_i, and $\sum n_i = n$.

In step (3) for each subsequence S_i, finding the minimum number and the maximum number needs the time complexity of $O(n_i)$, then the total time complexity is $\sum O(n_i)$, since $\sum n_i = n$, step (3) has the $O(n)$ time complexity.

Step (4) and step (5) need the time complexity of $O(n_{sub})$, since $n_{sub} \leqslant n$, the two steps also take time of $O(n)$.

Each step of the extraction procedure needs the time complexity that is linear to the amount of blocks, so the whole procedure takes the time complexity of $O(n)$. Then the theorem is proved.

4.2 Evaluation of Placement Regularity

The quality of a placement consists of topology quality and physical dimension quality.

4.2.1 Topological Structure Value

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

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

$$V_{top} = \alpha * \sum_{r \in R} \sigma(r) + \beta * \sum_{a \in A} \sigma(a), \tag{1}$$

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

4.2.2 Physical Dimension Cost

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

In a multi-row r, a block assigned to 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.

- Local compactness of a multi-row r,$C_{cmp}(r)$.

  If r is composed of horizontal single rows,

$$C_{cmp}(r) = \max_i \left(\sum_j w(r_{i,j}) \right) * \sum_i \max_j (h(r_{i,j})) - \sum_{i,j} a(r_{i,j}), \tag{2}$$

  Otherwise,

$$C_{cmp}(r) = \sum_j \max_i (w(r_{i,j})) * \max_j \left(\sum_i h(r_{i,j}) \right) - \sum_{i,j} a(r_{i,j}). \tag{3}$$

- Local uniformity of a multi-row r, $C_{uni}(r)$.

  If r is composed of horizontal single rows,

$$C_{uni}(r) = \sum_i (\max_j(w(r_{i,j})) - \min_j(w(r_{i,j}))), \tag{4}$$

Otherwise,

$$C_{uni}(r) = \sum_j (\max_i(h(r_{i,j})) - \min_i(h(r_{i,j}))). \tag{5}$$

The local compactness and the local uniformity as for arrays can be analogously defined, but they are omitted here for the space. Finally, we define the physical dimension cost C_{phy} as follows;

$$\begin{aligned} C_{phy} = \quad & \alpha' * \sum_{r \in R} (C_{cmp}(r) + C_{uni}(r)) \\ & + \beta' * \sum_{a \in A} (C_{cmp}(a) + C_{uni}(a)), \end{aligned} \tag{6}$$

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

4.3 Optimization Procedure

4.3.1 Dual Simulated Annealing

There is no limitation in usage of our 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 the structures of a placement. In this section, we propose a *dual simulated annealing* framework, and convey a constructive feature to it. Our original objectives are V_{top} and C_{phy}, that is, one is for the topology, the other is for the physical dimension. The *dual simulated annealing* 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(temperature, E, E1) then
10:         SP := SP1;
11:         E := E1;
12:         KeepBestSoFar(SP, E);
13:      end if
14:   end for
15:   P := GeneratePlacement(SP);
16:   E := E_phy(P);
     /* physical optimization */
17:   for (Physical structure searching) do
18:      SP1:= MovePhysicalStructure(SP);
19:      P := GeneratePlacement(SP1);
20:      E1 := E_phy(P);
21:      if IsAccept(temperature, E, E1) then
22:         SP := SP1;
23:         E := E1;
24:         KeepBestSoFar(SP, E);
25:      end if
26:   end for
27: end for
```

The cost function E_{top} and E_{phy} will be described in section 4.3.3.

4.3.2 Moves

In the above framework, the topology is optimized between line 5-14, while the physical dimension is done between line 17-26. We apply different moves to the different optimization, *MoveTopologicalStructure* and *MovePhysicalStructure*, they are explained as a combinations of FullExchange and HalfExchange introduced in[13]. For the completeness, we briefly describe them here.

- FullExchange: Pair interchange of two blocks in both Γ_+ and Γ_-.
- HalfExchange: Pair interchange of two blocks in either Γ_+ or Γ_-.

MoveTopologicalStructure: We execute *HalfExchange* on a sequence-pair to generate another sequence-pair. This move changes the sequence-pair and optimizes the topology.

MovePhysicalStructure: We apply *FullExchange* and *HalfExchangeKeepingStructure* to a sequence-pair. MovePhysicalStructure optimizes the physical dimension, meanwhile tries not to destroy the topology resulted from *MoveTopologicalStructure*. The operation of *HalfExchangeKeepingStructure* is interchange of two subsequences in both Γ_+ and Γ_-, it is designed to keep the inner topologies of multi-rows and arrays. Since a multi-row and an array are both rectangular extractable, each structure corresponds to a subsequence of the sequence-pair. Then, we apply a subsequence-exchange keeping blocks order inner the subsequences.

4.3.3 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. The chip area is calculated as the bounding box area of P, and the wire length is measured by the half-perimeter estimation. We describe about the primary objective in experiments, here it is denoted by $F(P)$. Then both functions, E_{top} and E_{phy}, are calculated as follows;

$$E_{top}(P) = \frac{F(P)}{g(V_{top})}, \tag{7}$$

$$E_{phy}(P) = F(P) * g(C_{phy}). \tag{8}$$

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

$$g(x) = 1.0 + 0.1 \exp\left(\frac{x_m \ln(0.5)}{x + \epsilon}\right), \tag{9}$$

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

5. Symmetry-Oriented Structured Placement

In analog circuits, partnered devices are often required to be placed symmetrically because of their sensitivities, matching the partnered devices helps to avoid both high offset voltage and degradation of power supply rejection ratio [6].We propose the symmetry-oriented structured placement making a placement naturally go to symmetrically.

5.1 Generation of Symmetry Structure

Firstly, we introduce the generation of symmetry structure without specifying any pair of devices or name of any block. Let a sequence-pair of n blocks be $\mathcal{SP} = (\Gamma_+, \Gamma_-)$, and let the reverse sequences of Γ_+ and Γ_- be Δ_+ and Δ_- respectively. $\Gamma_+(k)$ $(1 \leqslant k \leqslant n)$ and $\Delta_+(n-k+1)$ correspond to the same block as well as $\Gamma_-(k)$ and $\Delta_-(n-k+1)$ do to the same block. Originally, a sequence-pair is defined by the arrangement of blocks from the left-side to the right-side over the chip, we call such a sequence-pair *LR-SP*. On the other hand, if a sequence-pair is defined from the right-side to the left-side, the sequence-pair is called *RL-SP*. We notice that *RL-SP* is (Δ_-, Δ_+). If block x is the i-th in Δ_-, we denote $\Delta_-(i) = x$, as well as $\Delta_-^{-1}(x) = i$. Similar notation is also used for Δ_+. In a placement presented by *RL-SP*, for every block pair (a, b), a is left to b (equivalently, b is right to a) if $\Delta_-^{-1}(a) > \Delta_-^{-1}(b)$ and $\Delta_+^{-1}(a) > \Delta_+^{-1}(b)$. Analogously, a is below b (equivalently, b is above a) if $\Delta_-^{-1}(a) > \Delta_-^{-1}(b)$ and $\Delta_+^{-1}(a) < \Delta_+^{-1}(b)$. In the example of Fig. 5, *LR-SP* is $(\Gamma_+, \Gamma_-) = (b, d, a, c; a, b, c, d)$, while the RL-SP is $(\Delta_-, \Delta_+) = (d, c, b, a; c, a, d, b)$.

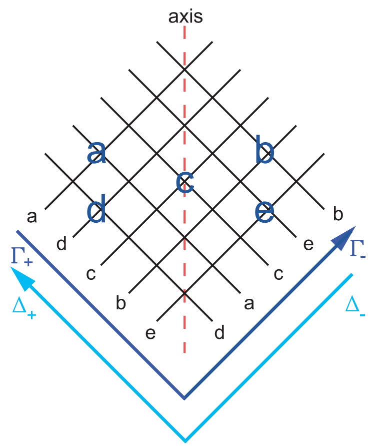


Fig. 7 An example of sequence-pair with horizontal symmetry topology.

The *RL-SP* can also induce a single-sequence, which is defined as $\mathcal{S}(k) = \Delta_-^{-1}(\Delta_+(k))$. If the *LR-SP* and the *RL-SP* induce the same single-sequence, that is,

$$\Gamma_+^{-1}(\Gamma_-(k)) = \Delta_-^{-1}(\Delta_+(k)) \ (1 \leqslant k \leqslant n), \tag{10}$$

then we say that the sequence-pair has a *horizontal symmetry topology*. **Figure 7** shows an sequence-pair with horizontal symmetry topology. The *LR-SP*$(\Gamma_+, \Gamma_-) = (adcbe, daceb)$ and the *RL-SP*$(\Delta_-, \Delta_+) = (becad, ebcda)$ introduce the same single-sequence which is (21354), so the sequence-pair has a *hori-*

Table 1 An example of single-sequence generation with a horizontal symmetry topology.

k	x	$n-k+1$	$n-x+1$	Γ_- and Δ_-	check-list
initial				(-, -, -, -, -)	(1, 2, 3, 4, 5)
1	2	-	-	(2, -, -, -, -)	(1, 3, 4, 5)
1	2	5	4	(2, -, -, 5, -)	(1, 3, 4)
2	1	-	-	(2, 1, -, 5, -)	(3, 4)
2	1	4	5	(2, 1, -, 5, 4)	(3)
3	3	-	-	(2, 1, 3, 5, 4)	(-)

zontal symmetry topology.

In the following, we introduce the generation of a single-sequence with an arbitrary horizontal symmetry topology. A horizontal symmetry topology has a single vertical axis. It can be also extended to a horizontal axis or plural axes.

(1) Make a check-list of $(1, 2, \ldots, n)$. Set Γ_+ as $(1, 2, \ldots, n)$, Δ_+ as $(n, n-1, \ldots, 1)$, Γ_- and Δ_- as empty, and $k = 1$.

(2) Stop the procedure if all the numbers have been assigned to Γ_-, that is, the check-list is empty.

(3) Assign an arbitrary number x in the check-list to $\Gamma_-(k)$, and remove x from the check-list. This step assigns a block to an arbitrary position.

(4) Assign $n-k+1$ to both $\Delta_-(x)$ and $\Gamma_-(n-x+1)$, and remove $n-k+1$ from the check-list. This step assigns another block into the symmetric position(symmetrical to the block assigned in step (3)).

(5) Increment k until $\Gamma_-(k)$ is not assigned yet, and return to step (2).

An example is shown in **Table 1** and **Fig. 8**. $\Gamma_-(k)$ and $\Delta_+(k)$ (or $\Gamma_+(k)$ and $\Delta_-(k)$) correspond to a symmetry-pair. Note that if they correspond to the same block, it means a self-symmetry.

Furthermore, in the table, $x = 2$ and $x = 1$ are chosen arbitrarily (see step (3) above). If we take other choices, the resultant Γ_- would be different. In other words, we can control the generation of a single-sequence with a horizontal symmetry topology by choosing these numbers.

Consider the implementation of a symmetrical placement from a sequence-pair with a horizontal symmetry topology. Prior works [1),2)] introduce a special calculation to place a pair of blocks imposed a symmetry-constraint on so that the y-axis becomes the center between them and their y-coordinates are the same [1),2)]. But, it suffers from the time-consuming calculation of y-coordinates,

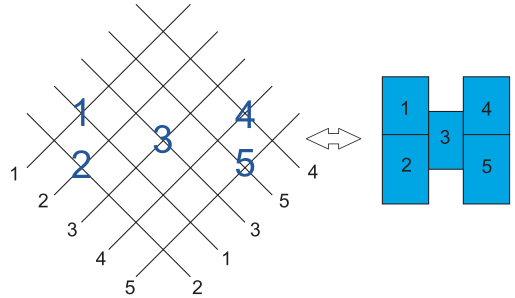


Fig. 8 An example of generation of symmetry structure.

because it often needs several iterations to align y-coordinates of a horizontal symmetry-pair. However, we can give a single path calculation of y-coordinates of symmetry-pairs as long as the sequence-pair has a horizontal symmetry topology.

In a placement with horizontal symmetry topology, if the y-coordinates of each symmetry-pair are same, the placement is called *vertical feasible*.

Theorem 5.1 Given an asyclic vertical constraint graph G_v induced by a sequence-pair with a horizontal symmetry topology, the calculation of all y-coordinates of a vertical feasible placement satisfying G_v with a horizontal symmetry topology, is linear in the time complexity to the edge number of G_v.

proof: Because the given sequence-pair has a horizontal symmetry topology, it is obvious that if blocks of each symmetry-pair have the same height, their y-coordinates are also the same after the longest path calculation of the vertical constraint graph.

Let each symmetry-pair be (b, b'). Give the maximum height of b and b' to both vertices corresponding to b and b'. Then, calculate the longest path from the grand source to each vertex. Note that the longest path calculation is linear to the number of edges. Therefore all the y-coordinates can be determined in the time complexity to the edge number of G_v during searching the longest path.

5.2 Initial Placement for Optimization

A normal simulated annealing procedure is adopted to execute the symmetry-oriented optimization, and it stars from a symmetrical initial placement. We generate a single-sequence satisfying a topological complete symmetry, then assign blocks into the single-sequence to generate the symmetrical initial placement. There are symmetry-pair numbers and self-symmetry numbers in the single-sequence of initial placement, we set a limitation on the amount of self-symmetry numbers in order to avoid too many self-symmetry blocks. If two numbers consist one symmetry-pair, one number calls the other *partner*. In this assignment, blocks are classified according to their size so that blocks of the similar size belong to the same group and the partner is chosen from the same group.

5.3 Moves

We propose new move operations to maintain the topological symmetry during the SA process. For the chosen blocks a and b, our moves are explained as a combination of FullExchange(a, b) and HalfExchange(a, b), same as mentioned before.

In the following, we describe all the moves used in the symmetry-oriented optimization. The simulated annealing controls the selection of moves according to random values. The partner of block a is denoted by a'. If a is a self-symmetry block, a' is equal to a.

- RotateBlock(a): Rotate the block a by 90 degree,
- FlipBlock(a): Flip the block a.
- FullExchangeOfSymm(a, b): Let the partners of a and b be a' and b', respectively. If a' is equal to b, or if a and b are both self-symmetry blocks, apply FullExchange(a, b). Otherwise, apply both FullExchange(a, b) and FullExchange(a', b'). Examples are shown in **Fig. 9**. Note in the Fig. 9 D, a is a self-symmetry block, i.e. a' is a. a and b are interchanged in both Γ_+ and Γ_-, then a' (a itself) and b' are interchanged similarly. This move will change the sequence-pair but not the topology of the placement, since the single-sequence is the same before and after the move.
- HalfExchangeOfSymm(a, b): Let the partners of a and b be a' and b', respectively. If a' is equal to b, or if a and b are both self-symmetry blocks, apply HalfExchange(a, b). Note that in this exchange, two self-symmetry blocks will be turned into a symmetry-pair, while a symmetry-pair will be separated into two self-symmetry blocks. Examples are shown in **Fig. 10** A and B. Otherwise if a and b are not partner for each other, apply HalfExchange(a, b) on Γ_+(or Γ_-) and HalfExchange(a', b') on Γ_-(or Γ_+). Examples are shown in Fig. 10 C and D. Note that in the Fig. 10 D, the self-symmetry block a and another block b are interchanged in Γ_+, then a' (a itself) and b' are interchanged in Γ_-. This move changes the placement's topology, symmetry partners are even separated into self-symmetry blocks like in Fig. 10 A, while our cautious measures keep the placement still topological symmetrical after the change.

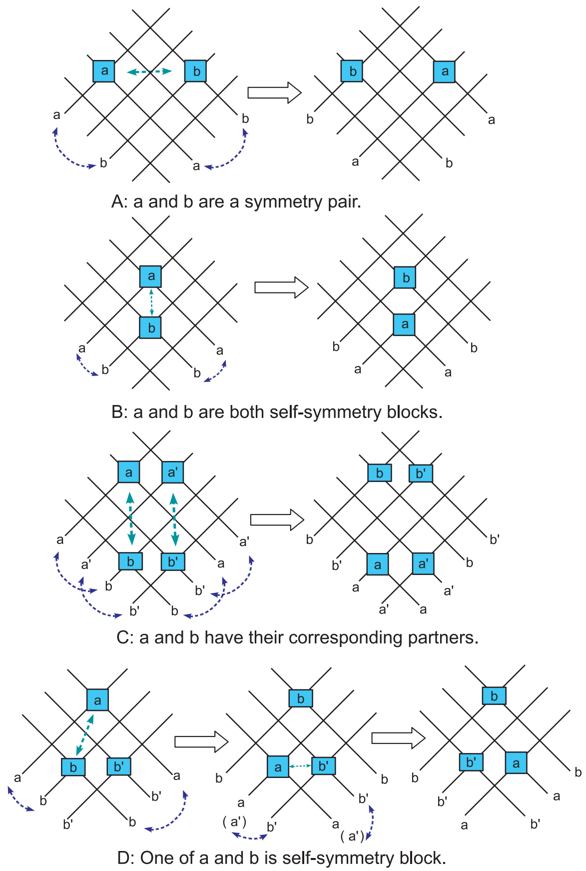


Fig. 9 Moves Keeping Symmetry: FullExchangeOfSymm on two blocks a and b.

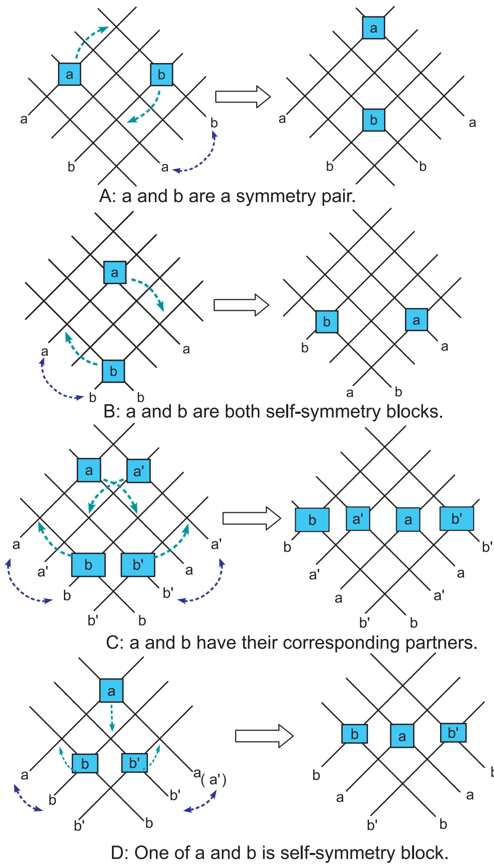


Fig. 10 Moves Keeping Symmetry: HalfExchangeOfSymm on two blocks a and b.

5.4 Cost Function

A symmetry structure can also be evaluated with respect to topology and physical dimension. Given a symmetry x, $|x|$ denotes the amount of total blocks and $|x_{self}|$ denotes the amount of self-symmetry blocks. b' denotes the partner of the block b. The topology quality of a symmetry x is valued as:

$$V_{top}(x) = |x| - |x_{self}|, \tag{11}$$

and we evaluate its physical dimension quality by defining the local symmetricity as:

$$C_{sym}(x) = \sum_{b,b' \in x} |w(b) - w(b')| * |h(b) - h(b')|. \tag{12}$$

Since the regularity is also preferred, we incorporate the evaluation of symmetry and regularity into the cost function as follows, so that not only do the resultant placements exhibit excellent symmetry, they show good regularity as well.

- The *Topological structure value*:

$$V_{top} = \alpha * \sum_{r \in R} \sigma(r) + \beta * \sum_{a \in A} \sigma(a) + \gamma * V_{top}(x) \tag{13}$$

  where the first two items are the same as in formula (1), α, β andγ are coefficients to balance a trade-off among structure values;
- The *Physical dimension cost*:

$$\begin{aligned} C_{phy} = \ & \alpha' * \sum_{r \in R} (C_{cmp}(r) + C_{uni}(r)) \\ & + \beta' * \sum_{a \in A} (C_{cmp}(a) + C_{uni}(a)) \\ & + \gamma' * \sum_{x \in X} C_{sym}(x), \end{aligned} \tag{14}$$

  where the first two items are the same as in formula (6), α', β' andγ' are coefficients to balance a trade-off among structure values;
- Then cost function used in symmetry-oriented SA procedure is designed as follows:

$$E(P) = \frac{Area(P) \cdot WLen(P)}{g(V_{top})} * g(C_{phy}), \tag{15}$$

  where $Area(P)$ and $WLen(P)$ are the chip area and the wire length of P respectively, g is the same as in formula (9).

5.5 Optimization under Constraints

Although our structured placement does not depend on constraint, the controllability of the initial placement makes it not difficult to combine the constraint-driven approaches with ours. For a specified pair of blocks which should be symmetrical, i.e., they need to satisfy one symmetry constraint, we assign them into a symmetry-pair and keep them as the partner of each other, the flip and rotate operations will be executed on both blocks. Since the placement's topol-

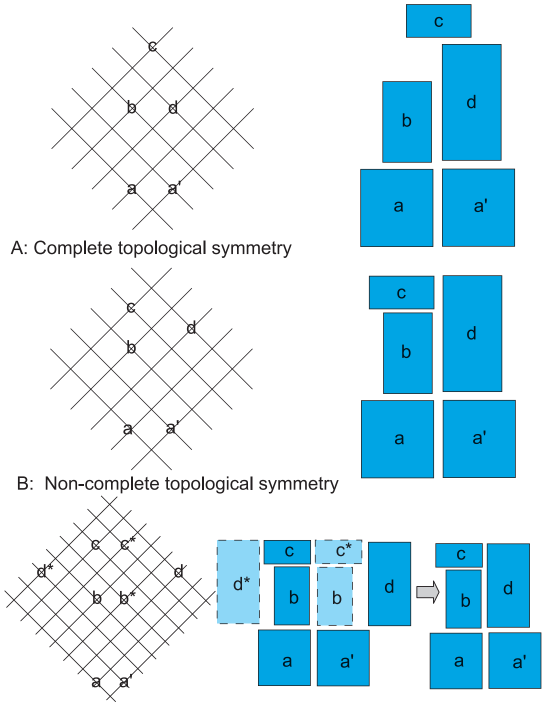


Fig. 11 Physical skewed symmetry structure.

ogy may change because of the moves, block and its partner might be separated into self-symmetry blocks, blocks even exchange their partners with other blocks. The moves that may violet the constraints (such as Fig. 9 D) are not allowed for the blocks bonding with constraints. Then the specified blocks satisfy the constraints, putting other blocks symmetrically helps to thermal distribution and symmetrical routing, meanwhile regularity is also preserved.

5.6 Physical Skewed Symmetry Structure

Because of the limitation of the block's size or shape, a complete symmetry placement may lead to too much compromise on chip area and wire length. **Figure 11** shows an example with 5 blocks: a, a', b, c, and d. a and a' have the same size and shape. A complete symmetry-oriented placer trends to get a placement like Fig. 11 A, which is complete topological symmetry, but might be not so good at the area usage. And there could be another solution like Fig. 11 B, which can not be obtained by a complete symmetry placement, but the placement still looks symmetrically because the area of it's left side and right side are almost same. In order to make improvement on such kind of topological complete symmetry placement, we insert some dummy blocks into the single-sequence, and the size of each dummy block is 0. In a symmetry structure, if a block's symmetry partner is a dummy block, we say that they are formulated as a *Physical Skewed Symmetry Structure*.

During the generation of the initial placement and the optimization, the single-sequence keeps being topological symmetry. The dummy blocks are ignored in the symmetry coordinate calculation, so it is possible to generate a physical skewed symmetry placement including non-symmetry parts. As a result we can get a more compacted placement. In Fig. 11 C, dummy blocks b^*, c^* and d^* are inserted into the single-sequence, and form a topological complete symmetry placement, finally can get the result like Fig. 11 B.

6. Experiments

6.1 Structured Placement

We tested our structured placement tool for analog block designs. We prepared 13 instances of analog block sets from industries. The amount of blocks and nets of each data are shown in **Table 2**. 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 SA. We compared the results with respect to the chip area, the wire length, and the structure coverage. The structure coverage is the ratio of the number of blocks composing topological arrays or rows to the total number of blocks.

First, we set the primary objective as the chip area. The numerical data of the results is shown in Table 2. For all instances, the ratio of the area by our structured placement to that by the normal placement is less than 1.05, while the structure coverage of the normal placement and that of our structured placement are quite different, 7.9% and 78.4% on the average respectively. The resultant

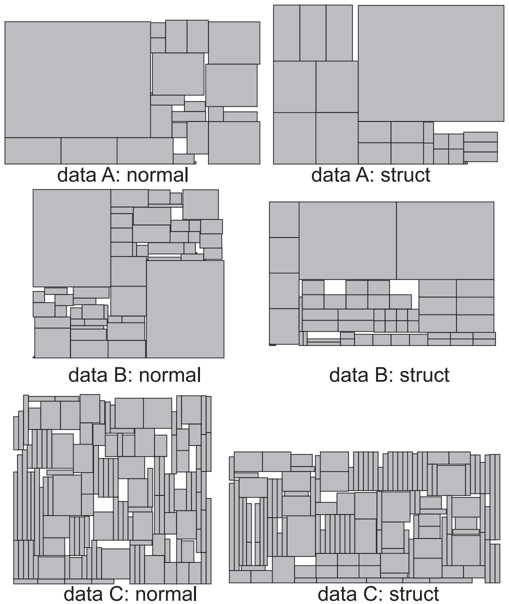


Fig. 12 Resultant placements of data A, B, and C, where the primary objective is the chip area: "normal" and "struct" are normal placement and our structured placement, respectively. It can be seen that the routing channels in the structured results are simpler and more regular.

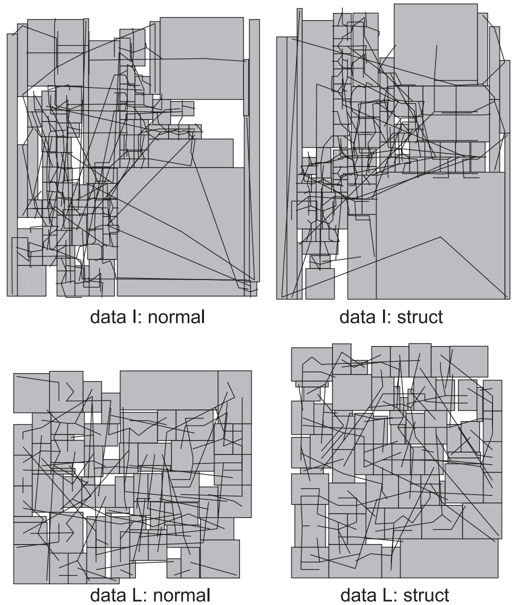


Fig. 13 Resultant placements of data I and L, where the primary objective is the product of the chip area and the wire length.

placements of data A, B, and C by both tools are shown in **Fig. 12**.

Second, we set the primary objective as the product of the chip area and the wire length. The numerical data of the results are shown in **Table 3**. Compared the results by our placement to those by normal placement, the ratio of the product of the chip area and the wire length is 0.978 on the average. The introduced regularity barely deteriorate the placement quality of chip area and wire length. 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. 13**.

Our method can obtain as many as possible regular structures, which conduce to simple and regular routing channels. These results can be observed from Fig. 12 and Fig. 13. For example, for data A in Fig. 12, it clearly shows that the routing channels in the structured placements are simpler and more regular than those in the normal placement. Using these channels, the net bends during the routing stage can be reduced, consequently introducing less vias and improving the performance by reducing the RC delay of nets.

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 enables a solution to escape from a local minimum. An observation of the similar kind is seen in the cluster-constraint-driven approach. However, the pair-exchanging of structures in our method does not depend on any constraint. Actually no matter if the placement is of constraint or not of constraint, those blocks under no constraint in the placement are automatically grouped into multi-rows or arrays, and the multi-rows or arrays are manipulated as single objects. That is why although our structured placement did not use any cluster constraint, the

Table 2 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 of the number of blocks composing topological arrays or rows to the total number of blocks . "area ratio" is the ratio of the area by "struct" to that by "normal".

data	# blocks	# nets	normal		struct		area ratio
			area (μm^2)	str-cover (%)	area (μm^2)	str-cover (%)	struct/normal
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
Average				7.9		78.4	1.00

Table 3 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 of the area*wire-len by "struct" to that by "normal".

data	normal			struct			area*wire-len
	area (μm^2)	wire-len (μm)	str-cover (%)	area(μm^2)	wire-len(μm)	str-cover (%)	ratio struct/normal
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
Average			9.6			73.5	0.978

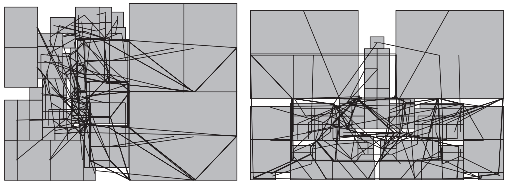


Fig. 14 Resultant placements of data O, where the primary objective is the product of the chip area and the wire length: "normal" and "symmetry" are normal placement and out symmetry placement, respectively.

similar effect can be observed.

6.2 Symmetry-Oriented Structured Placement

We also implemented a symmetry-oriented structured placement, and performed it on 14 instances.

6.2.1 Comparison with normal placement

Firstly, we compare our symmetry-oriented structured placement with normal one. **Figure 14** shows a pair of resultant placements. The results of normal placement are well compact, but not well organized in local area, and lack symmetry. Our symmetry-oriented structured placement is well compact either and distinguished for the perfect symmetry even no constraint is given. Since we combined the evaluation of row and array structures, our placement also shows good quality on local compactness and regularity. Our placement is strictly limited to be symmetrical, but the chip area and the wire length are not necessarily increasing heavily due to this limitation. The numerical results are shown in **Table 4**. The normal placement and our placement are denoted by *normal* and *symm*, respectively. The table includes the results with respect to the chip area and the wire length by both placement. The results show, as we expected, our placement of a complete symmetry structure could be obtained by the compromise of at most about 5 % on average, compared to the normal placement. Note that our symmetry-oriented structured placement is very fast, this owes to the topological symmetry enabling the simple calculation of vertical coordinate.

6.2.2 Physical Skewed Symmetry

Secondly, we demonstrate the effectiveness of *physical skewed symmetry* structure. For the same instances, we generated the dummy blocks as the 10%, 20%, and 30% of the amount of input blocks, and tested each case. The numerical test results are shown in **Table 5**. In this table, values to chip area and wire length are ratios to the chip area and wire length by *symm* in Table 4. The columns of "Best" show the best result among those of "dummy 0%", "dummy 10%", "dummy 20%" and "dummy 30%". We attained the 6% and 14% reduction on average with respect to chip area and wire length. **Figure 15** shows the results by *normal*, *complete symmetry* and *physical skewed symmetry*. We are con-

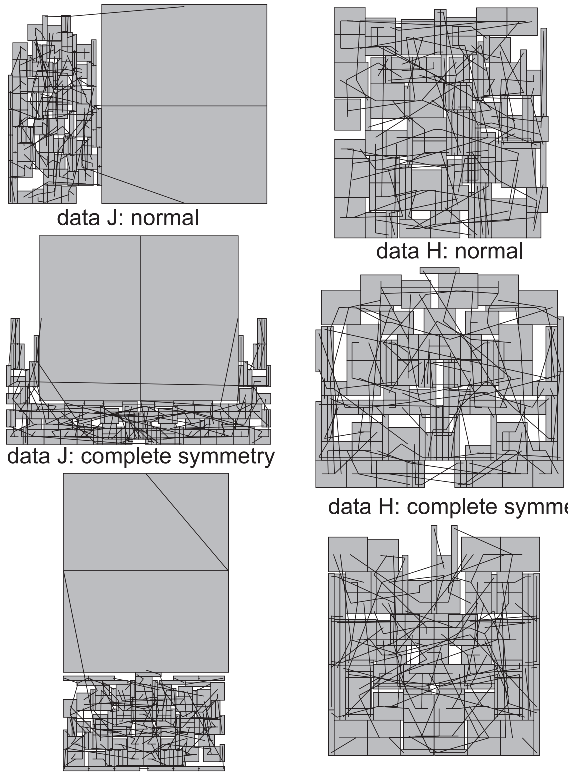


Fig. 15 Resultant placements of data J and H, where "normal", "complete symmetry" and "physical skewed symmetry" are normal placement, our complete symmetry placement and physical skewed symmetry placement respectively.

Table 4 The numerical result of analog block designs, where"wlen" is the wire length, "normal" and "symmetry" refer to normal placement and our symmetry placement respectively.

data	#	#	normal			symm			(symm-normal) /symm	
	blocks	nets	area(μm^2)	wlen(μm)	time(s)	area(μm^2)	wlen(μm)	time(s)	area (%)	wlen (%)
C	122	91	102,660	9,356	1,581	107,940	10,444	840	5.14	11.63
D	60	46	80,784	5,391	241	100,082	7,007	221	23.89	29.98
E	113	80	265,068	16,520	1,261	257,420	14,820	684	-2.89	-10.29
F	32	22	69,090	2,413	49	74,844	3,120	75	8.33	29.30
G	54	49	67,077	3,286	172	71,760	3,023	177	6.98	-8.01
H	90	58	94,105	9,680	700	103,968	9,786	497	10.48	1.10
I	60	36	12,683	2,465	236	13,008	2,530	238	2.56	2.61
J	101	78	42,452	3,392	1,868	45,456	4,692	1,148	7.08	38.31
K	66	29	173,631	4,124	632	193,953	3,699	534	11.70	-10.30
L	64	49	58,622	5,274	263	60,543	6,043	258	3.28	14.58
M	166	105	75,507	15,576	7,926	80,565	12,645	1,600	6.70	-18.82
N	60	44	159,799	3,715	463	163,681	3,010	352	2.43	-18.97
O	55	91	865,860	38,209	376	916,491	36,717	449	5.85	-3.91
P	22	53	543,753	8,390	29	497,151	10,107	56	-8.57	20.46
average:									5.93	5.55

Table 5 The numerical result of analog block designs using *Physical Skewed Symmetry Structure*.

data	dummy 0%		dummy 10%			dummy 20%			dummy 30%			Best(0–30%)	
	area (%)	wlen (%)	#du-mmy	area (%)	wlen (%)	#du-mmy	area (%)	wlen (%)	#du-mmy	area (%)	wlen (%)	area (%)	wlen (%)
C	100	100	12	91.2	111.7	24	102.3	106.6	36	98.6	113.3	91.2	100
D	100	100	6	91.1	75.7	12	91.2	83.2	18	85.5	81.7	85.0	75.7
E	100	100	11	124.9	104.7	22	125.5	98.4	33	114.2	99.8	100	98.4
F	100	100	3	96.9	130.4	6	98.3	103.9	9	96.4	83.3	96.4	83.3
G	100	100	5	109.0	97.3	10	99.2	108.5	16	98.6	108.7	98.6	97.3
H	100	100	9	89.3	103.3	18	95.9	96.6	27	94.4	96.4	89.3	96.4
I	100	100	6	84.2	80.6	12	90.3	82.6	18	96.5	76.0	84.2	76.0
J	100	100	10	89.8	81.4	20	90.6	85.3	30	89.2	75.9	89.2	75.9
K	100	100	6	112.5	84.4	13	100.8	85.4	19	99.4	91.8	99.4	84.4
L	100	100	6	99.8	98.5	12	100.8	95.4	19	102.5	106.7	99.8	95.4
M	100	100	16	102.2	107.3	33	98.6	73.2	49	99.6	94.7	98.6	73.2
N	100	100	6	94.0	141.3	12	98.7	118.5	18	101.7	109.4	94.0	100.0
O	100	100	5	98.2	93.2	11	129.5	153.9	16	96.2	105.3	96.2	93.2
P	100	100	2	106.1	113.9	4	104.6	63.9	6	100.2	73.2	100.0	63.9
average				99.2	101.7		101.9	96.8		98.0	94.0	94.4	86.6

vinced that the insertion of the dummy blocks expands the solution space of the symmetry-oriented structured placement, and helps to reduce the chip area and wire length.

7. Conclusion

We introduced a new concept–structured placement, which makes use of the regularity as a key criterion. The regularity is formulated in terms of topological structures such as arrays and rows. We provided the extraction of the regular structures from a single-sequence in $O(n)$, as well as the way to evaluate the structures. Also, we proposed a new SA framework, called dual simulated annealing, where we convey a constructive feature to an SA framework so as to optimize placements' topological and physical properties separately and alternately. Besides, we presented a symmetry-oriented structured placement which can generate symmetrical placement with good regularity.

In experiments for 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. The experiments also demonstrate the effectiveness of our symmetry-oriented structured placement.

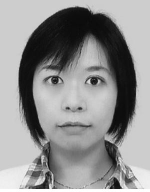
We believe that there is no limitation in usage of our structured placement in general placement and floorplanning framework. Actually, we are convinced that it is necessary to combine the constraint-driven method with our structured placement to develop a more practical tool. In future works, we will apply the structured placement to analog floorplanning along with further practical extensions.

References

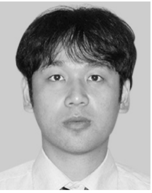
1) Balasa, B. and Lampaert, K.: Symmetry within the sequence-pair representation in the context of placement for analog design, *IEEE Trans. on CAD*, Vol.19, No.7, pp.721–731 (2000).
2) Balasa, F., Maruvada, S.C. and Krishnamoorthy, K.: On the exploration of the solution space in analog placement with symmetry constraints, *IEEE Trans. on CAD*, Vol.23, No.2, pp.177–191 (2004).
3) Chang, Y.-C., Chang, Y.-W. and Wu, G.-M.: B*-trees: A new representation for non-slicing floorplan, *Proc. 37th DAC*, pp.458–463 (2000).
4) Dong, Q. and Nakatake, S.: Constraint-free Analog Placement with Topological Symmetry Structure, *Proc. ASP-DAC 2008*, pp.186–191 (2008).
5) Guo, P.N., Cheng, C.K. and Yoshimura, T.: An O-tree representation of non-slicing floorplan and its applications, *Proc. 36th DAC*, pp.267–273 (1999).
6) Cohn, J., Garrod, D., Rutenbar, R. and Carley, L. : *Analog Device-level Automation*, Kluwer Acad. Publi. (1994).
7) Kodama, C. and Fujiyoshi, K.: Selected Sequence-Pair: An efficient decodable packing representation in linear time using Sequence-Pair, *Proc. ASP-DAC 2003*, pp.331–337 (2003).
8) Lin, J. and Young, Y.: TCG-S: Orthogonal coupling of P*-admissible representations for general flooplans, *Proc. 39th DAC*, pp.842–847 (2002).
9) Lin, J.M. and Chang, Y.W.: TCG-S: Orthogonal coupling of P*-admissible representations for general floorplans, *IEEE Trans. on CAD*, Vol.24, No.6, pp.968–980 (2004).
10) Lin, J.-M. and Chang, Y.-W.: A transitive closure graph based representation for general floorplans, *IEEE Trans. on VLSI Systems*, Vol.13, No.2, pp.288–292 (2005).
11) Lin, P.-H. and Lin, S.-C.: Analog placement based on novel symmetry-island formulation, *Proc. 39th DAC*, pp.465–470 (2007).
12) Murata, H., Fujiyoshi, K., Nakatake, S. and Kajitani, Y.: Rectangle-packing-based module placement, *Proc. ICCAD 1995*, pp.472–479 (1995).
13) Murata, H., Nakatake, S., Fujiyoshi, K. and Kajitani, Y.: VLSI module placement based on rectangle-packing by Sequence-Pair, *IEEE Trans. on CAD*, Vol.15, No.12, pp.1518–1524 (1996).
14) Nakatake, S.: Structured Placement with Topological Regularity Evaluation, *Proc. ASP-DAC 2007*, pp.215–220 (2007).
15) Nakatake, S., Fujiyoshi, K., Murata, H. and Kajitani, Y.: Module packing based on the BSG-structure and IC layout applications, *IEEE Trans. on CAD*, Vol.17, No.6, pp.519–530 (1998).
16) Tam, Y.-C., Young, E.F. and Chu, C.: Analog placement with symmetry and other placement constraints, *Proc. ICCAD 2006*, pp.349–354 (2006).
17) Zhang, X. and Kajitani, Y.: Space-planning: Placement and modules with controlled empty area by Single-Sequence, *Proc. ASP-DAC 2004*, pp.25–30 (2004).

(Received November 17, 2008)
(Revised February 20, 2009)
(Accepted April 13, 2009)
(Released August 14, 2009)

(Recommended by Associate Editor: *Masaya Yoshikawa*)



Qing Dong received the B.S. degree in computer science and the M.S. degrees in computer software and theory from University of Electronic Science and Technology of China, China, in 2001 and 2004, respectively. She had been a researcher of R&D Center for Semiconductor Design Environment of the Kitakyushu University, Japan, from 2005 to 2006. Since April 2006, she has been working towards the doctoral degree at the University of Kitakyushu. Her research interests include simulation algorithm and VLSI layout design.



Shigetoshi Nakatake received the B.E. degree from Tokyo Institute of Technology in 1992, the M.S. degree from Japan Advanced Institute of Science and Technology in 1994, and the D.E. degree from Tokyo Institute of Technology in 1999. He had been a research associate in department of electrical and electronic engineering of Tokyo Institute Technology from 1996 to 1999. Since 1999, he has been an associate professor of the University of Kitakyushu. His research interests are mainly focused on physical design and mixed signal design.