A Method of Analog IC Placement with Common Centroid Constraints

Keitaro Ue Kunihiro Fujiyoshi
Department of Electrical and Electronic Engineering
Tokyo University of Agriculture and Technology
2-24-16 Nakachou, Koganei-shi, Tokyo 184-8588, Japan
keitaro@fjlab.ei.tuat.ac.jp, fujiyosi@cc.tuat.ac.jp

Abstract—To improve the immunity against process gradients, a common centroid constraint, in which every pair of capacitors which has been derived by dividing some original capacitors into two halves should be placed symmetrically with respect to a common centroid, is widely used. Xiao et al. proposed a method to obtain a placement satisfying the common centroid constraints, but this method has a defect. In this paper, we propose a decoding algorithm to obtain a placement which satisfies common centroid constraints.

I. INTRODUCTION

Recently most of integrated circuit chips are made from monolithic IC, which can be manufactured at low cost. In monolithic IC, absolute error which is proportion of the difference of actual value to the designed value of each element is large (several tens of percentages), but the maximum difference of the absolute error of devices on a chip, is small (a few percentages). So it is suitable to build circuits such as A/D, D/A converters and switched-capacitor filters. But if the accuracy of these circuits is required to be higher, such a small mismatch of capacitors or resistors still becomes a critical issue. The causes of the mismatch can be divided into two categories: systematic mismatch and random mismatch [1, 2]. One of the frequent systematic mismatch is caused by process gradients, whose effect can be represented by the following linear model [3]:

\[ C_{xy} = C + \alpha x + \beta y, \]

where \( C_{xy} \) is the capacitance of the unit capacitor in the coordinates \( x \) and \( y \), \( C \) is the ideal value of the unit capacitance, and \( \alpha \) and \( \beta \) are the process gradient components in \( x \) and \( y \) directions, respectively.

Therefore to improve the immunity against process gradients, a common centroid constraint, in which every pair of capacitors which has been derived by dividing some original capacitors into two halves should be placed symmetrically with respect to a common centroid, is widely used.

In order to obtain placement which satisfies common centroid constraint, some methods are proposed [4, 5, 6, 7, 8]. Strasser et al. proposed a method using B*-tree [4], but this method is deterministic. P. H. Lin et al. proposed a thermal-driven common centroid placement algorithm [5], and C. W. Lin et al. proposed a method to obtain placement considering random mismatches [6]. Both methods [5, 6] can handle unit-capacitors only.

Two methods using sequence-pair (seq-pair) [9], which is a topological representation of rectangular block placement, are proposed [7, 8]. Ma et al. proposed a method using Center-based Corner Block List (C-CBL) [7], which is an extension of CBL [10]. They regard each common centroid group as a rectangular super-block, and search whole placements by seq-pair, so area obtained by the method may be redundantly large. Xiao et al. proposed a method which uses seq-pair to obtain placement satisfying common centroid constraint and searches seq-pairs which satisfy some constraints [8]. But their method has a defect that some blocks may be overlapped (Fig.1(e)).

In this paper, we introduce a method proposed in [8] and a defect of the method. And we propose a decoding algorithm of seq-pair to obtain a placement which satisfies common centroid constraints, without overlapping. The proposed decoding algorithm takes \( O(n^2) \) time. The effectiveness of the method will be examined by computational experiments.

II. PROBLEM DEFINITION

A common centroid constraint gives a constraint, in which every pair of cells in the same given group should be placed symmetrically with respect to a common centroid of the group. These groups are called common centroid groups. A common centroid group consists of some pairs of cells and at most one self-symmetry cell, and the center of the self-symmetry cell conforms to the common centroid of the group. A common centroid group is represented by a set of cell pairs in parentheses and at most one self-symmetry cell such as \( \{(a_1, a_2), b_s, (c_1, c_2)\} \).

In this paper, it is supposed that input seq-pair satisfies group clustered placement condition and common centroid condition, which are defined in section III., and cells in each common centroid group should be placed as close as possible to the common centroid. \( a_x \) denotes rel-
ative $x$-coordinate of left side of cell $a$, and $a_y$ denotes relative $y$-coordinate of bottom side of cell $a$. $a.height$ and $a.width$ denote height and width of cell $a$ respectively.

III. Method proposed in [8]

A method proposed by Xiao et al. searches only seq-pairs satisfying group clustered placement condition, which is the same as separate constraint defined in [12], and common centroid condition.

**common centroid condition:**

Suppose a seq-pair $S = \{G_i; \Gamma_i\}$ which includes common centroid group $G^1_i$ and $(\Gamma_{G_{i+1}}; \Gamma_{G_{i-1}})$ which is a sub-seq-pair obtained by extracting just the cells in $G^1_i$ from $S$. If $\Gamma_{G_{i+1}}$ and $\Gamma_{G_{i-1}}$ satisfy the following conditions, seq-pair $S$ satisfies common centroid condition.

$$\Gamma_{G_{i+1}} = \text{sym}(\text{rev}(\Gamma_{G_{i+1}}))$$
$$\Gamma_{G_{i-1}} = \text{sym}(\text{rev}(\Gamma_{G_{i-1}}))$$

where $\text{rev}(s)$ is a string obtained by reversing a string $s$ and $\text{sym}(s)$ is a string obtained by replacing a string $s$ by their symmetric counterparts.

The method in [8] has a defect that some cells may be overlapped. An example is as follows.

Suppose a seq-pair $S = (a_1eb_1a_2c_1d_1f_1; c_2e_1g_1d_2), \{a_2c_2\}$ is given, and common centroid groups are $G^1_i\{(a_1, a_2), b_2\}$, $G^2_i\{(c_1, c_2), d_1\}$. The size of $a_1$, $a_2$ and $b_2$ is $1 \times 1$, the size of $c_1$, $c_2$ and $d_1$ is $2 \times 1$, the size of $e$ and $g$ is $2 \times 2$, the size of $f$ is $1 \times 1.5$.

First, construct constraint graphs from input seq-pair $S$ and calculate coordinates of each cell by the longest path length from source to the corresponding vertex (Fig.1(a),(b)). Then cells in $G^1_i$ don’t satisfy common centroid constraint. To satisfy common centroid constraint, calculate potential displacement $\delta$ which $a_2$ should be shifted by (Algorithm 1 AdjustingHorizontalSymmetry in [8]). And add the $\delta$ to weights of all out-going edges of $a_2$ (Fig.1(c),(d)). Finally, as cells in $G^1_i$ and $G^2_i$ satisfy common centroid constraint, re-calculate coordinates of cells in $G^1_i$ and $G^2_i$. Then resultant placement is shown in Fig.1(e).

Fig. 1. Example of Algorithm AdjustingHorizontalSymmetry proposed in [8], which results in the overlap of cell $d_2$ and $f$.

IV. Proposed Method

In symmetric placement, to determine the relative coordinates of pairs of cells which should be placed symmetrically with respect to a symmetrical axis is possible by using longest path algorithm on a constraint graph, if these cells should be placed as close as possible to the axis [11]. For example, suppose a pair of cells $(a_1$ and $a_2$) in a symmetric group is constrained horizontally by seq-pair, and $a_1$ is constrained to be left of $a_2$. The relative $x$-coordinates of $a_1$ and $a_2$ satisfy the following relations:

$$\text{subject to } a_1.x + a_1.width + a_2.x = 0$$
$$a_2.x \geq 0$$

minimize $a_2.x$.

where relative $x$-coordinate of the symmetric axis is zero. So these relative $x$-coordinates are not difficult to determine. And relative $y$-coordinates are not difficult to determine because the only additional constraint is to align relative $y$-coordinates of cells in each pair.

In common centroid placement, if $a_1$ is constrained to be left of $a_2$, the relative $x$-coordinates of these cells are not difficult to determine by the same reason. But relative $y$-coordinates of these cells are difficult to determine, because the only relation of the relative $y$-coordinates of cells in the pair is

$$a_1.y + a_1.height + a_2.y = 0.$$
placed in the third quadrant, cells in $B_i$ are placed in the fourth quadrant (Fig.2).

The fundamental idea has the following weak points:

(1) If $|S_i| \neq 0$, there exist dead space in each quadrant around the only element of $S_i$ as shown in hatched region on Fig.5(j).

(2) If $|S_i| = |A_i| = |B_i| = 0$, vertical distance between cells in $R_i$ and cells in $L_i$ is redundantly large as shown in Fig.3(a).

(3) If $|S_i| = |R_i| = |L_i| = 0$, horizontal distance between cells in $A_i$ and cells in $B_i$ are redundantly large.

### B. proposed decoding algorithm

To overcome the above weak point (1), we change the position of the following four regions in which cells in each subset $R_i, A_i, L_i, B_i$ are placed (Fig.4). If $|S_i| = |\{s\}$,

- **if** $a \in R_i$ : \[
\begin{align*}
&\frac{s \text{.width}}{2} \leq a \cdot x \\
&\frac{s \text{.height}}{2} \leq a \cdot y 
\end{align*}
\]

- **if** $a \in A_i$ : \[
\begin{align*}
&s \text{.width} \\
&\frac{s \text{.height}}{2} \leq a \cdot y + a \cdot \text{width} 
\end{align*}
\]

- **if** $a \in L_i$ : \[
\begin{align*}
&\frac{-s \text{.width}}{2} \leq a \cdot x + a \cdot \text{width} \\
&\frac{s \text{.height}}{2} \geq a \cdot y 
\end{align*}
\]

- **if** $a \in B_i$ : \[
\begin{align*}
&\frac{s \text{.width}}{2} \leq a \cdot x \\
&\frac{-s \text{.height}}{2} \geq a \cdot y + a \cdot \text{height} 
\end{align*}
\]

To overcome the weak point (2), if $|S_i| = |A_i| = |B_i| = 0$, we change the relative $y$-coordinates of cells in $L_i$, and to overcome the weak point (3), if $|S_i| = |R_i| = |L_i| = 0$, we change the relative $x$-coordinates of cells in $A_i$. Proposed decoding algorithm is as follows.

**step1**: Based on the horizontal (left of) constraints imposed by the seq-pair, a directed weighted graph $G_h(V, E)$ ($V$: vertex set, $E$: edge set), called horizontal constraint graph is constructed as follows [9].

- $V$ consists of source $sv$ and vertices labeled with the corresponding names of all cells.

- $E$ consists of $(sv, p)$ for every cell $p$, and $(p, q)$ if and only if $p$ is constrained to be left of $q$ by the seq-pair.

- Length of edge $(p, q)$ is the width of the cell $p$.

Similarly, the vertical constraint graph $G_v$ is constructed using vertical (below) constraints and the height of each cell.

**step2**: For each common centroid group $G^i$, carry out the following steps.

If $|S_i| \neq 0$, suppose the only self-symmetry cell in $G^i$ is denoted as $s$. If $|S_i| = 0$, $s$.

**step2.1**: Make induced subgraph $G^i_h$ of $G_h$, induced by vertices in $R_i$, $B_i$, and $S_i$. Add source $sv$ and edge from $sv$ to every vertex to $G_h^i$. For vertex $c$,

- **weight**$(sv, c) = \begin{cases} 
0 : (\text{if } c \in R_i) \\
-\frac{s\text{.width}}{2} : (\text{if } c \in (B_i \cup S_i)) 
\end{cases}$

Calculate the longest path length from $sv$ to each vertex and regard them as relative $x$-coordinates of left side of all cells in $G^i$.

**step2.2**: Make induced subgraph $G^i_v$ of $G_v$, induced by vertices in $R_i$, $A_i$, and $S_i$, and add source $sv$ and edges $(sv, c)$, similar to step2.1. For vertex $c$,

- **weight**$(sv, c) = \begin{cases} 
0 : (\text{if } c \in A_i) \\
-\frac{s\text{.height}}{2} : (\text{if } c \in (R_i \cup S_i)) 
\end{cases}$

Calculate the longest path length from $sv$ to each vertex and regard them as relative $y$-coordinates of bottom side of all cells in $G^i$.

**step2.3**: If $|S_i| = |A_i| = |B_i| = 0$:

Change the relative $y$-coordinates of cells in $L_i$ so that the minimum relative $y$-coordinate of cells in $R_i$ is equal to the minimum relative $y$-coordinate of cells whose relative $x$-coordinate of right side is equal to zero in $L_i$. (See Fig.3(b)).

**step2.4**: If $|S_i| = |R_i| = |L_i| = 0$:

Change the relative $x$-coordinates of cells in $A_i$ so that the minimum relative $x$-coordinate of cells in $B_i$ is equal to the minimum relative $x$-coordinate of cells whose relative $y$-coordinate of bottom side is equal to zero in $A_i$.

Regard a set of cells in each symmetry group as one convex block.
**step3:** Coordinates of all cells are determined by convex block packing algorithm, proposed in [12]. Detailed operations are as follows.

**step3.1:** Adjust weight of edges in $G_h$ using relative coordinates obtained in step2. Weight of edge $(a, b)$ is adjusted to

$$a.width + a.x - b.x,$$

where $a.x$ is relative $x$-coordinate of $a$ in $G^i$ if $a \in G^i$, and $a.x = 0$ if $a$ is an asymmetry cell, $b.x$ is similarly.

Similarly adjust weight of edges in $G_v$.

**step3.2:** Merge all the vertices in $G^i$ into one “super-vertex”, and calculate the longest path length from source to all vertices in $G_h$ and $G_v$. And let these longest path lengths be $x$ and $y$ coordinates of corresponding cells or convex blocks.

The time complexity of step1 and step2 is obviously $O(n^2)$, and that of step3 is $O(n^2)$, which is shown in [12]. So the total time complexity of this proposed decoding algorithm is $O(n^2)$.

For example, suppose a seq-pair $(b_1 a_1 c_1 a_2 b_2 d_1 b_2 d a_1 c_1 a_2 b_1)$ which satisfies common centroid condition and group clustered placement condition is given. And common centroid group is $G^1 \{ (a_1, a_2), (b_1, b_2), c_1 \}$. The size of $a_1, a_2$ and $c_1$ is $1 \times 1$, the size of $b_1, b_2$ and $d$ is $1 \times 2$.

step1: constraint graphs $G_h, G_v$ of the input seq-pair are shown in Fig.5(a), (b).

step2.1: induced subgraph $G^1_h$ of $G_h$ induced by cells in the symmetry group $G^1$ is shown in Fig.5(c). step2.2: induced subgraph $G^1_v$ of $G_v$ induced by cells is shown in Fig.5(d).

step3.1: for each constraint graph, widths of edges are adjusted using relative coordinates obtained in step2 (Fig.5(e),(f)). step3.2: merge all the vertices in $G^1$ into one “super-vertex” as shown in Fig.5(g),(h). Calculate the longest path length from source to all vertices in $G^1_h$, and let these values be $x$-coordinates of corresponding cells or convex blocks. Similarly calculate the longest path length from source to all vertices in $G^1_v$, and let these values be $y$-coordinates of that (Fig.5(i)). For comparison, placement which is obtained from the same input by method of fundamental idea is shown in Fig.5(j).

**V. EXPERIMENTAL RESULT**

In order to confirm effectiveness of the proposed decoding algorithm, we implemented the decoding algorithm with a method of our fundamental idea with Simulated Annealing (SA) in C language, and carried out experiments by Athlon 2.7GHz for two data used in [4] and [8], which are “biasynth_2p4g” and “lnamixbias_2p4g”. “biasynth_2p4g” consists of 65 cells with 3 common centroid groups (4 pairs, 6 pairs, and 2 pairs and 1 self symmetric cell). “lnamixbias_2p4g” consists of 110 cells with five common centroid groups (8 pairs, 3 pairs, 3 pairs, 6 pairs, and 2 pairs). Table I shows the experimental comparisons of results of the proposed method with [8]. “Area ratio” is defined by the following equation:

$$\text{Area ratio} = \frac{(\text{area of bounding box})}{(\text{total area of all cells})}.$$ 

We cannot simply compare these results, because we considered all groups in “biasynth_2p4g” and “lnamixbias_2p4g” are common centroid groups, but [8] considered two groups in “biasynth_2p4g” and 3 groups in “lnamixbias_2p4g”. The area of placement obtained by the proposed decoding algorithm is less than that by the method of fundamental idea, though the CPU time is almost similar. The packing results of “biasynth_2p4g” obtained by proposed decoding algorithm and that by the method of fundamental idea are shown in Fig.6 and Fig.8, and the packing results of “biasynth_2p4g” are shown in Fig.7 and Fig.9 respectively.

**VI. CONCLUSION**

In this paper, we proposed a decoding algorithm to obtain placement satisfying common centroid constraints and relative position constraints imposed by seq-pair, which satisfies both groups clustered placement condition and common centroid condition, in $O(n^2)$ time. And, in experiments, feasible placements satisfying common centroid constraints were obtained in about two hundred seconds.

**References**


TABLE I

<table>
<thead>
<tr>
<th>Design</th>
<th>#Cell</th>
<th>#Symmetry groups</th>
<th>method in [8]</th>
<th>fundamental idea</th>
<th>proposed decoding algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>biasynth_2p4g</td>
<td>65</td>
<td>8+12+5</td>
<td>18.2</td>
<td>103.91</td>
<td>152.0 110.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>141.6 108.88</td>
</tr>
<tr>
<td>lnamixbias_2p4g</td>
<td>110</td>
<td>16+6+6+12+4</td>
<td>58.7</td>
<td>104.31</td>
<td>229.0 116.56</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>222.4 112.39</td>
</tr>
</tbody>
</table>

Fig. 5. An example of the decoding algorithm: common centroid group $G_1 = \{(a_1,a_2),(b_1,b_2),cs\}$, asymmetry cell $d$, seq-pair $S = (b_1a_1c_1a_2b_2d; b_2d a_1c_1a_2b_1)$
Fig. 6. Resultant placement of “biasynth_2p4g” obtained by method of fundamental idea (Time: 152.0[sec], Area ratio: 110.43[%])

Fig. 7. Resultant placement of “lnamixbias_2p4g” obtained by method of fundamental idea (Time: 229.0[sec], Area ratio: 116.56[%])

Fig. 8. Resultant placement of “biasynth_2p4g” obtained by proposed decoding algorithm (Time: 141.6[sec], Area ratio: 108.88[%])

Fig. 9. Resultant placement of “lnamixbias_2p4g” obtained by proposed decoding algorithm (Time: 222.4[sec], Area ratio: 112.39[%])