

Explicit Expression and Simultaneous Optimization of Placement and Routing for Analog IC Layouts

Yukiko KUBO [†], Shigetoshi NAKATAKE [†], Yoji KAJITANI [†], and Masahiro KAWAKITA [‡]

[†]Dept. of Information and Media Sciences, The University of Kitakyushu

[‡]System LSI Div., Semiconductor Company, Toshiba Corp.

{kubo, nakatake, kajitani}@env.kitakyu-u.ac.jp, masahiro.kawakita@toshiba.co.jp

Abstract

Our target is automation of analog circuit's layout, which is a bottleneck in mixed-signal's design. We formulate the layout explicitly considering manufacturing process, and propose an algorithm that consists of simultaneous expression and optimization of placement and routing. The key is that all the cells and wires are represented by rectangles. The algorithm is combined into a commercial tool, and the performance convinced us that the utilization shortens the design time.

1 Introduction

Recently, demand of wireless instruments becomes rapidly growing, so we are being requested to make progress in design productivity of analog circuits e.g. RF modules. A system scale LSI called SoC (System on a Chip) becomes including not only a digital circuit but also such an analog circuit, that is, almost of types of SoCs come to mixed-signals.

Generally, a scale of the analog part is small, while the design period is long. It is that manual process is still dominant through the design flow, and it is the only way to empirically design by experts. The features of analog circuits are that the performance is very sensitive to noise or other electrical factors, and that it is also close to the manufacturing process. These cause complicated constraints in the layout design [1, 2]. We face to an issue that a layout by any automation tool cannot overcome a manual design. Because the manual design simultaneously handles placement and routing, the quality is sophisticated in considering the constraints.

However, we note an available feature that the scale is comparatively small, for example, one module consists of a few hundreds of cells and nets. Therefore, we can handle carefully the design rules and specifications in both placement and routing.

This paper provides a break-through such that placement and routing are executed simultaneously. This brings us an approach close to a manual design. [4] also proposes an algorithm of simultaneous placement and routing which uses the BSG, but it is far from practical design rules of analog circuits.

Our algorithm explicitly takes the design rules into consideration. It is based on the Sequence-Pair algorithm (denoted by SPa) which is a rectangle placement[3]. A key of our idea is in representing wires by a sequence-pair as well as cells. Each wire is divided into a set of rectangles, and the following two extensions are introduced in order to maintain the connection; One is to impose a condition of orders of rectangles on a sequence-pair, and the condition is called Wire-connectivity. The other is to generate horizontal and vertical

constraint graphs for compaction. We add special edges to the conventional constraint graphs which are extracted from a sequence-pair.

We lead a theorem that our expression of a sequence-pair with Wire-connectivity is not redundant (i.e. optimal) with respect to area. We can seek an optimal placement and routing on the expression. Thus, we attain the way to simultaneously place and route.

We complete our idea by implementation. We embedded our algorithm into a commercial tool for analog IC layouts. The performance was sufficed for design specification and computation time. A comment of the designer is that utilization of the tool shorten design time for 30-50%. Furthermore, we demonstrate that the algorithm is flexible to the existing techniques [5, 6, 7] based on SPa, which are to handle coordinates-fixed or symmetric cells. This means that our algorithm can optimize placement and routing simultaneously under such constraints.

2 Shape-Based Formulation

2.1 Multiple Outline Circuitry Model

In order to explicitly handle design rules of analog circuits, we introduce *Multiple Outline Circuitry Model*. First, we define cells which correspond to transistors, resistors, condensers, diodes, vias and so on. A set of cells is denoted by $C = \{c_1, c_2, \dots, c_n\}$. The cell consists of multiple outlines which depend on manufacturing process, and the outlines are rectangles which correspond to layers. A terminal is considered as one of outlines. A set of layers is denoted by $L = \{l_1, l_2, \dots, l_k\}$, and a set of the outlines in a cell c is denoted by $O(c) = \{o_1, o_2, \dots\}$. The width and height of an outline o_i are denoted by $w(o_i)$ and $h(o_i)$, respectively. The layer to which o_i corresponds is denoted by $l(o_i) (\in L)$. A cell c has x- and y-coordinate, which are $x(c)$ and $y(c)$, respectively. o_i belonging to c has relative coordinates to c , regarding $(x(c), y(c))$ as the origin. Accordingly, the coordinates of o_i on the plane are $(x(c) + x(o_i), y(c) + y(o_i))$.

Next, we define wires. A set of wires is denoted by $\Pi = \{\pi_1, \pi_2, \dots, \pi_m\}$. A wire connects two or more terminals. When it connects two terminals, it has a sequence of coordinates. The sequence is called path and denoted by $p = (x_1, y_1)(x_2, y_2)(x_3, y_3) \dots$. The coordinates correspond to a corner or an end on the path, and the length of p is the number of the corners and ends. A path consists of vertical and horizontal segments, where if $x_i = x_{i+1}$ then $y_i \neq y_{i+1}$, otherwise $y_i = y_{i+1}$. A wire may connect three or more terminals, and ends of a path may lie not at a terminal but in the middle

of another path. Such an end is called Steiner point. A wire consists of a set of paths, that is $P(\pi) = \{p_1(\pi), p_2(\pi), \dots\}$. A path p belongs to one of layers denoted by $l(p)$, and it has the width denoted by $w(p)$. Thus, a wire represented by a set of paths is called *path-based wire*.

We are also given a design rule, which is a set of distances to any pair of layers in order to make an edge of a layer apart from that of the other layer. These distances are defined horizontally and vertically according to manufacturing process. The minimal vertical and horizontal distances for layers l_i and l_j are denoted $d_v(l_i, l_j)$ and $d_h(l_i, l_j)$, respectively. The design rule is kept if $d \geq d_h(l_i, l_j)$ or $d' \geq d_v(l_i, l_j)$, where d and d' are the horizontal and vertical distances between edges of l_i and l_j , respectively.

Then, our target problem is defined as follows.

General Shaped-Based Layout Problem

input: C and Π
output: the coordinates of C and Π
(i.e. placement and routing)

subject to:

design rules with respect to C and Π are kept

objective:

minimize area of the minimal rectangle (bounding box)
enclosing C and Π

2.2 Rectangular Dissection of Wire

To handle wires and cells simultaneously in layout automation, we introduce *rectangle-based wire model*, where a wire is represented by a set of rectangles. Each rectangle is called *wire-rect*.

Let a path be p . The corresponding set of wire-rects is denoted by $R(p) = \{r_1(p), r_2(p), \dots\}$. Each wire-rect r has the coordinates, width, and height, denoted by $(x(r), y(r), w(r), h(r))$, respectively.

The set of wire-rects to p is obtained as follows.

1. Let two terminals of p be t_a and t_b , respectively.
2. Divide p into segments and put the order to them along p from t_a to t_b .
3. For each segment $(x_i, y_i)(x_j, y_j)$, expand it into such a rectangle that
 - (a) the left-bottom corner is $(\min(x_i, x_j) - w(p)/2, \min(y_i, y_j) - w(p)/2)$,
 - (b) the width is $x_i - x_j$ if the segment is horizontal, $w(p)$ otherwise, and
 - (c) the height is $x_i - x_j$ if the segment is horizontal, $w(p)$ otherwise.
4. Put the same order to each rectangle that of the segment, and let the sequence of rectangles be $\{r_1, r_2, \dots\}$.
5. Extract area where r_i overlaps with r_{i+1} , and delete the area.
6. Extract area where r_i overlaps with t_a and t_b , and delete the area.

An example of the procedure is described in the following and in Figure 1. Assume that p has two ends (x_1, y_1) , (x_4, y_4) , and two corners (x_2, y_2) , (x_3, y_3) . Let terminals connected with p at (x_1, y_1) and (x_4, y_4) be t_a and t_b , respectively. t_a and t_b belong to the same layer as that of p . Let the coordinates, width, and height of t_a and t_b be (x_a, y_a, w_a, h_a) and (x_b, y_b, w_b, h_b) , respectively. Let three segments along p be s_1 ,

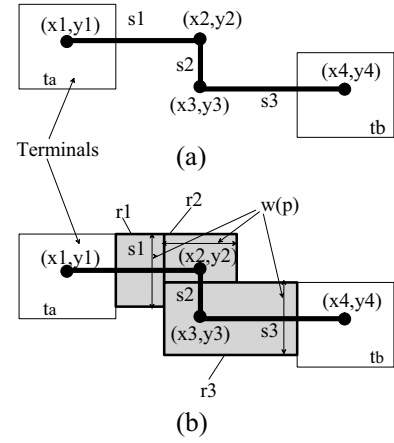


Figure 1: Rectangle-based wire model

s_2 , and s_3 , where s_1 is horizontal ($s_1 = (x_1, y_1)(x_2, y_2)$, ($x_1 < x_2$)), s_2 is vertical ($s_2 = (x_2, y_2)(x_3, y_3)$, ($y_2 < y_3$)), and s_3 is horizontal ($s_3 = (x_3, y_3)(x_4, y_4)$, ($x_3 < x_4$)). Let three wire-rects be r_1 , r_2 , and r_3 , and they are $(x_a + w_a, y_1 - w(p)/2, x_2 - w(p)/2 - x_a, w(p))$, $(x_2 - w(p)/2, y_2 - w(p)/2, w(p), y_2 - y_3)$, and $(x_3 - w(p)/2, y_3 - w(p)/2, x_3 - w(p)/2 - x_b, w(p))$, respectively.

Hence, we can obtain a set of wire-rects which corresponds to any path, where any pair of wire-rects does not overlap as well as any wire-rect does not overlap with any terminal.

The above procedure is available in both cases of three or more terminals and multiple layers by applying simple extensions.

3 Simultaneous Expression of Placement and Routing

Our inputs which consist of cells and rectangle-based wires are all sets of rectangles. We provide an extension of *the Sequence-Pair* [3] which is an algorithm to place rectangles into as small area. Hereinafter the Sequence-Pair is denoted by SPa.

For SPa, an input and an output are a set of rectangles and the placement without overlapping, respectively. The objective is to minimize the area of bounding box of the rectangles. The algorithm uses a pair of sequences of rectangle's name called sequence-pair which represents a placement. A sequence-pair is denoted by (α, β) , where α and β are sequences of names of rectangles. α is denoted by $a_1 a_2 \dots$, and β is by $b_1 b_2 \dots$.

In applying SPa, for simplicity of description, we assume that all the wire-rects belong to the same layer. An extension to multiple layers is described later.

In our definition of a sequence-pair, α and β are sequences of names of cells and wire-rects. The length of each sequence is :

$$|C| + \sum_{\pi \in \Pi} \sum_{p \in P(\pi)} \{(\text{length of } p) - 1\} .$$

The order in α (β) with respect to a cell or a wire-rect is given by a reverse function α^{-1} (β^{-1}).

3.1 Wire-Connectivity

We have to keep the adjacency such that a wire is retrieved by connecting the wire-rects after compaction. We

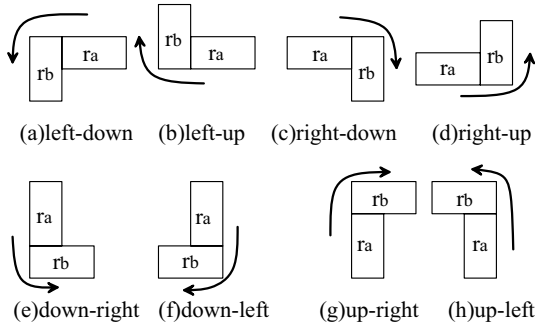


Figure 2: 8 patterns of an adjacent pair of wire-rects

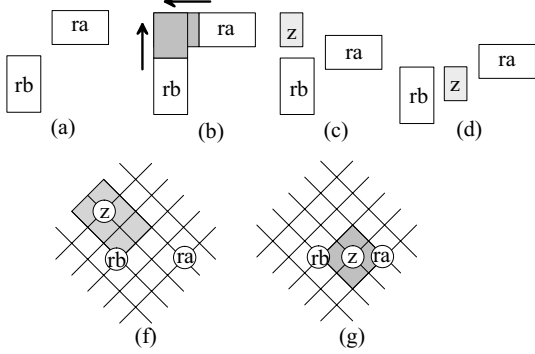


Figure 3: J-connectivity

introduce a condition to keep the adjacency, which is defined on a sequence-pair. The condition is called *Wire-connectivity*, and it consists of sub-conditions *J-connectivity* (at jog), *T-connectivity* (at terminal), and *S-connectivity* (at Steiner point).

J-Connectivity: This condition guarantees retrieval with respect to a pair of wire-rects in a path. Let an adjacent pair of wire-rects at a corner of a path be r_a and r_b where $a < b$. The number of patterns for connection of r_a and r_b is eight. These patterns are shown in Figure 2.

In the case of (a)left-down in the figure, we describe the condition for r_a and r_b on a sequence-pair. Assume that, after compaction, r_a and r_b are placed as shown in Figure 3(a). We retrieve the connection of r_a and r_b as in (b), but the other rectangle z may interrupt the retrieval, as shown in (c) and (d). Then (f) and (g) show sequence-pairs represented by the oblique line grids. If z lies in the shadow regions, then z may interrupt the retrieval. Otherwise z does not.

Thus, if r_a , r_b , and any other rectangle z are put in a sequence-pair such that $\beta^{-1}(z) < \beta^{-1}(r_b)$, $\beta^{-1}(z) > \beta^{-1}(r_a)$, or $\alpha^{-1}(z) > \alpha^{-1}(r_a)$ then z does not interrupt in the retrieval procedure.

Analogously, in the other patterns in Figure 2, condition for r_a , r_b , and z on a sequence-pair are obtained. We omit them here for the space.

T-Connectivity: We provide the condition for retrieval with respect to a terminal and a wire-rect. Let a terminal and the connecting wire-rect be t and r , respectively. After compaction, t and r are placed as shown in Figure 4(a). To retrieve the connection as shown in (b), the other rectangle z is as in (c) may interrupt the retrieval. If z does not lie in the shadow regions in (d), z does not interrupt. Therefore,

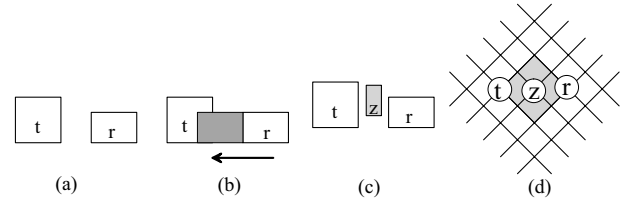


Figure 4: T-connectivity

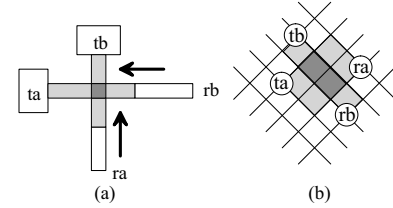


Figure 5: A conflict of retrieval

condition is: $\beta^{-1}(z) < \beta^{-1}(t)$, $\beta^{-1}(z) > \beta^{-1}(r)$, $\alpha^{-1}(z) < \alpha^{-1}(t)$, or $\alpha^{-1}(z) > \alpha^{-1}(r)$.

Furthermore, we have to prevent retrieval from conflicting with respect to two terminals as shown in Figure 5(a) (The sequence-pair is shown in (b)). The figure shows the conflict of two connections between terminals and wire-rects. Thus, the sequence-pair must satisfy $\alpha^{-1}(r_b) < \alpha^{-1}(t_a)$, $\alpha^{-1}(r_a) < \alpha^{-1}(t_b)$, $\beta^{-1}(t_b) < \beta^{-1}(r_a)$, or $\beta^{-1}(r_a) < \beta^{-1}(r_b)$.

We also have to keep a similar condition for retrieval of terminal-to-wire-rect not to conflict with that of terminal-to-terminal.

S-Connectivity: Let two wire-rects connected at a Steiner point be r_a and r_b . As the discussion of T-connectivity, in retrieving the connection of r_a and r_b , condition for the other rectangle z not to interrupt are as follows. $\beta^{-1}(z) < \beta^{-1}(r_a)$, $\beta^{-1}(z) > \beta^{-1}(r_b)$, $\alpha^{-1}(z) < \alpha^{-1}(r_a)$, or $\alpha^{-1}(z) > \alpha^{-1}(r_b)$.

Furthermore, as well as T-connectivity, we have to keep a condition to prevent retrieval from conflicting with the other.

3.2 Extension of Compaction under Wire-Connectivity

Given a sequence-pair which meets Wire-connectivity, we try to get the corresponding placement and routing by compaction. In order to guarantee connections after the compaction, we introduce additional directed edges to G_h and G_v . The compaction is called *WC-compaction*.

In our problem, cells consist of several outlines (rectangles). We provide a pre-procedure before compaction to calculate the minimal horizontal and vertical distances between cells c_a and c_b . These distances are denoted by $d_h(c_a, c_b)$, $d_v(c_a, c_b)$ ($c_a, c_b \in E$, $a \neq b$), respectively.

$$d_h(c_a, c_b) = x(c_b) - x(c_a) = \max_{\forall o_i \in c_a, \forall o_j \in c_b} \{x(o_i) + w(o_i) + d_h(l(o_i), l(o_j)) - x(o_j)\}$$

$$d_v(c_a, c_b) = y(c_b) - y(c_a) = \max_{\forall o_i \in c_a, \forall o_j \in c_b} \{y(o_i) + h(o_i) + d_v(l(o_i), l(o_j)) - y(o_j)\}$$

An example is shown in Figure 6(a). The horizontal distance between two cells c and c' is $d_h(c, c') = x(o_2) + w(o_2) + d_h(l(o_2), l(o_4)) - x(o_4)$.

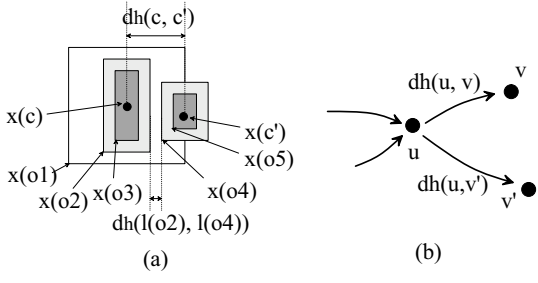


Figure 6: The distance between c and c' and weighting directed edges

As for the distances between cell c and a wire-rect r , they are to be calculated in the following two cases. One is the distances from c to r , which are $d_h(c, r)$ and $d_v(c, r)$. $d_h(c, r) = x(r) - x(c) = \max_{\forall o_i \in c} \{x(o_i) + w(o_i) + d\}$, where if o_i is not a terminal connecting with r then $d = 0$, otherwise $d = d_h(l(o_i), l(r))$. $d_v(c, r) = y(r) - y(c) = \max_{\forall o_i \in c} \{y(o_i) + h(o_i) + d\}$, where if o_i does not connect with r then $d = 0$, otherwise $d = d_v(l(o_i), l(r))$.

The other is the distances from r to c . $d_h(r, c) = x(c) - x(r) = w' + \max_{\forall o_i \in c} \{d - x(o_i)\}$ where if o_i is not a terminal connecting with r then $d = 0$, otherwise $d = d_h(l(r), l(o_i))$, and if r corresponds to a horizontal segment then $w' = 0$, otherwise $w' = w(r)$. $d_v(r, c) = y(c) - y(r) = h' + \max_{\forall o_i \in c} \{d - y(o_i)\}$ where if o_i does not connect with r then $d = 0$, otherwise $d = d_v(l(o_i), l(r))$, and if r corresponds to a vertical segment then $h' = 0$, otherwise $h' = h(r)$.

Similarly, the distances between two wire-rects r_i and r_j which are $d_h(r_i, r_j)$ and $d_v(r_i, r_j)$ are calculated, but we omit the ways here.

In [3], each vertex of G_v corresponds to a rectangle and is given to the width of the rectangle as a weight. And each vertex of G_v also corresponds to a rectangle and is given to the height.

In this paper, we give not a vertex but each directed edge the distance which is between two rectangles connected by the edge. An example of weighting in G_h is shown in Figure 6(b). Furthermore, we introduce additional directed edges to keep Wire connectivity. These edges are categorized into three types of *J-edge*(at terminal), *T-edge*(at terminal), and *S-edge*(at Steiner point). For simplicity, we describe such additional edges by using examples in the following.

J-Edge: To keep connection as shown Figure 7(A)(a), the width of r_2 must be positive, that is, $x(r_1) \leq x(r_3)$ must hold. Then we add an edge to G_v , which is from r_1 to r_3 and given the weight 0 (See Figure 7(A)(b)).

T-Edge: To retrieve connection of a wire-rect and a terminal, relative distance between the wire-rect and the cell to which the terminal belongs must be maintained after compaction.

In Figure 7(B)(a), distance between cell c and wire-rect r is d . We add two edges; (i) one is from r to c with the weight of $-d$. (ii) the other is from c to r with the weight of d . (See Figure 7(B)(b)).

Note that to add those two edges means that G_h becomes directed cyclic. However, when there exist a placement and a routing which corresponds to the sequence-pair, the sum of

weights of edges on the cycle is not positive. Hence we can calculate the longest path length of G_h in polynomial time.

S-Edge: As shown in Figure 7(C), paths p_1 and p_2 are connected at the Steiner point on r_2 . In this case, we add an edge from r_1 to r_4 with the weight 0.

This constraint may be too severe. We already have how to cancel it. To apply that way, however, we need complex discussion, so omit it here.

Constructing G_h and G_v taking the extensions of J-, T-, and S-edges, coordinates of cells and wires-rects are determined as follows; For each cell c (each wire-rect r), $x(c)$ ($x(r)$) is the longest path length in G_h from the source to the vertex corresponding to c (r). Similarly, $y(c)$ ($y(r)$) is determined by G_v .

Corners and ends on a path p are determined according to the coordinates of wire-rects and terminals.

1. Let each of segments into which p is divided be s , and let a wire-rect corresponds to s be r .
2. If s is horizontal, y-coordinates of both ends of s are $y(r) + w(p)/2$. Otherwise, x-coordinates of both ends of s are $x(r) + w(p)/2$.
3. Select one in the following cases whether an end of s is to be connected with a terminal t and wire-rect $r' (\neq r)$.
 - (a) If the end is to be connected with t at right- or left-side of t , x-coordinate of the end is $x(c) + x(t) + w(t)/2$ where c is a cell to which t belongs.
 - (b) If the end is to be connected with t at top- or bottom-side of t , y-coordinate of the end is $y(c) + y(t) + h(t)/2$.
 - (c) If the end is to be connected with a vertical r' , x-coordinate of the end is $x(r') + w(r')/2$.
 - (d) If the end is to be connected with a horizontal r' , y-coordinate of the end is $y(r') + h(r')/2$.
4. Make each path p so thick that the width of p is $w(p)$.

Through the above operation (WC-compaction), we get the x- and y-coordinates of all the cells and wires without any design rule errors.

4 Optimality of Expression and Extension to Multiple Layer Routing

Our simultaneous expression brings us the following significant theorem.

Theorem 1 *For General Shaped-Based Layout Problem, every solution X can be mapped to a sequence-pair (α, β) which satisfies Wire-connectivity, and WC-compaction $((\alpha, \beta))$ outputs a solution with not more than the area of X .*

We complete the proof of the theorem, but we omit it here for the space.

Furthermore, we release an assumption that a layer of every wire is the same. Let the number of layers for the routing be k' , and these are denoted by $met_1, met_2, \dots, met_{k'}$. We prepare $k' + 1$ sequence-pairs, and these are denoted by $sp_1, sp_2, \dots, sp_{k'+1}$. We define each sequence-pair sp_i such that sp_i consists of wire-rects and cells which include met_i . sp_i satisfies Wire-connectivity. $sp_{k'+1}$ consists of cells which does not include any layers for the routing.

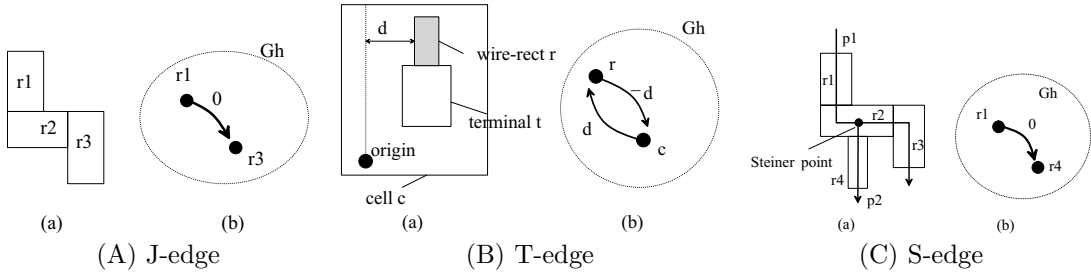


Figure 7: Examples of additional edges to G_h

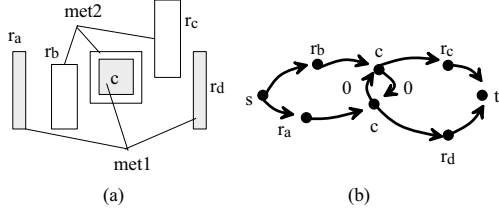


Figure 8: G_h for multiple layer routing

Next, let a horizontal constraint graph given by sp_i be $G_h(sp_i)$, to which is added J-, T-, and S-edges. For a cell c which includes both met_i and met_j ($i \neq j$), let a vertex to c be v in $G_h(sp_i)$, and let a vertex to c be v' in $G_h(sp_j)$. Add a pair of edges with a weight of 0 such that one is from v to v' and the other is from v' to v . Collect sources of all the graphs to one, and collect sinks to one.

An example is shown in Figure 8. The cell c belongs to met_1 and met_2 . The wire-rects r_a and r_b belong to met_1 , and r_b and r_c belong to met_2 . Two vertices to the cell c are connected by a pair of directed edges. As the results, we obtain G_h which is combined those graphs from $G_h(sp_1)$ to $G_h(sp_{k'})$. Analogously, G_v is obtained. The procedure of WC-compaction is the same. Significantly, Theorem 1 holds as for thus multiple sequence-pairs and WC-compaction.

5 Optimization of Placement and Routing

Our framework of the optimization is based on an iterative improvement strategy, as well as SPa.

1. Construct an arbitrary solution X , that is, a placement and a routing without any violation of the design rule.
2. Extract a sequence-pair (α, β) from X which keeps Wire-connectivity.
3. Repeat the following steps until a terminative condition is satisfied.
 - (a) Apply WC-compaction to (α, β) to create a new solution X' .
 - (b) Evaluate area of the bounding box of X' and keep the best solution so far.
 - (c) Change the sequence-pair by a move to create a new sequence-pair (α', β') keeping feasibility, where feasibility means that Wire-connectivity is satisfied and there is no positive cycle in G_h and G_h .
 - (d) $(\alpha, \beta) = (\alpha', \beta')$.

WS	Sun Ultra SparcII 360MHz
OS	Solaris 2.6
Domain EDA	Virtuso (Cadence Design Systems)
Engine	Coded by C-Language
I/F & GUI	Coded by SKILL
Design	Leaf-Cells on Bipolar

Table 1: Platform of our interactive tool

4. Output the best solution so far.

In the framework, an initial solution is given by the existing shape-based algorithm, although the performance is not enough. The move is very simple, and it is described as follows.

1. Select a cell or wire-rect and let it be x .
2. Select either sequence of α or β .
3. Remove x from the sequence (α or β).
4. Insert x into the sequence so as to keep feasibility.

Furthermore, we have to make it clear how to apply a move keeping feasibility. The details are omitted here. An example of two moves is shown in Figure 9. In the figure, (a) is an initial solution, and (c) is the corresponding sequence-pair. (d) shows that g is removed from β and inserted it between a and d of β . (e) shows that g is removed from α , and inserted it behind d of α . (b) is the placement and routing which corresponds to (e).

There does not exist a placement and a routing by infinite iterations of these type moves, so we need different types of moves. Note that a topology of each wire does not change during the iterations. We already have several ideas to change the topology, but we will introduce them in the next paper.

The algorithm is flexible to other extensions based on SPa, because our expression and WC-compaction are fundamental as well as SPa. For examples, we can adopt the following extensions; to handle coordinates-fixed cells (not to move) [5], to handle symmetric cells [7], and to handle rectilinear cells [6]. Note that these techniques are applicable to wires since we represent a wire by a set of rectangles.

6 Application to Analog IC Layouts

To complete our ideas, we implemented our algorithm embedded into a commercial tool. The platform is described in Table 1.

Our target designs are analog leaf-cells and blocks. An example of the performance of our engine is shown in Figure 10 and 11. Figure 10 is an initial entry. The algorithm searches

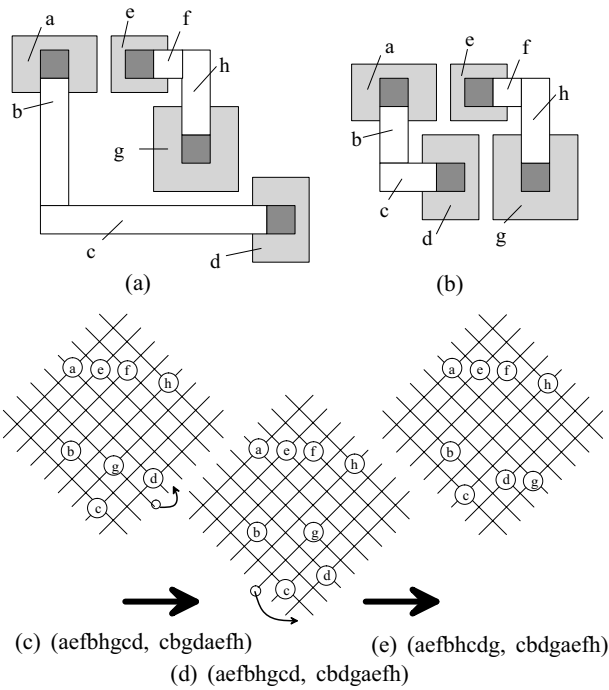


Figure 9: An example of moves on a sequence-pair

calc.time(sec)	0.300
no. of trials	2400
no. of feasible solutions	542

Table 2: The computation report of simultaneous placement and routing

the best solution by iteration of moves. The result is shown in Figure 11. We imposed symmetric constraints to five pairs of cells and a pair of wires. Note that we can attain symmetric wires by using the algorithm in [7], since a wire is represented by a set of rectangles.

The computation report is described in Table 2. A feasible solution means that the corresponding sequence-pair meets Wire-connectivity.

The layout is compacted, and our optimization process was executed very quickly, although the number of cells is small. A comment of a designer is that utilization of the tool enables him to shorten design time for 30-50%. The result can convince us that the tool also has sufficient potential to full-automation design of analog circuit layout.

7 Concluding Remarks

We aimed to shorten analog circuit design time which is a bottle-neck in the whole design period. We formulated the layout problem, and proposed an algorithm to solve it, which is based on SPa. The features of the proposals are: (i) multiple outline circuitry model, (ii) simultaneous expression on a sequence-pair by rectangle-based wire model, (iii) condition on a sequence-pair to guarantee connection of wires (iv) compaction under wire's connectivity, and (v) simultaneous optimization of placement and routing. We also led a theorem that our expression is not redundant (i.e. optimal) with respect to area. Furthermore, we demonstrated the performance of the engine embedded on a commercial tools. The

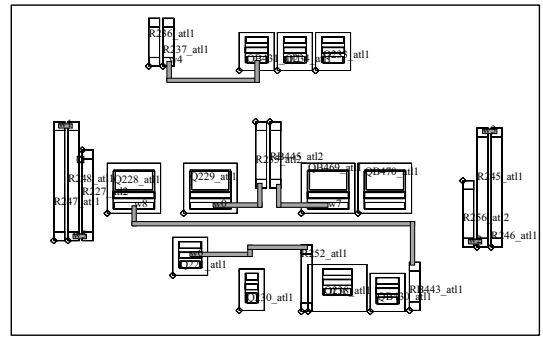


Figure 10: An initial placement and routing

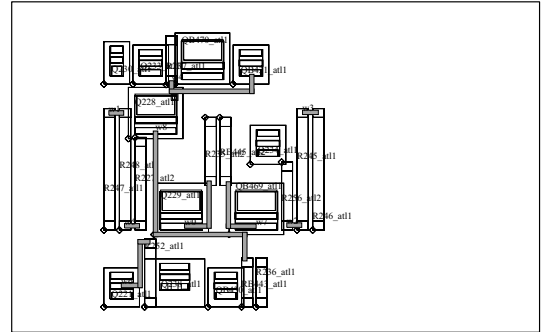


Figure 11: The result by simultaneous optimization

results convinced us that utilization of our tool shortens the design time. The algorithm is flexible to other techniques based on SPa, which are to handle symmetric, coordinates-fixed, and rectilinear cells. These techniques are also applicable to wires since we represent a wire by a set of rectangles.

References

- [1] J.Cohn, D.Garrod, R.Rutenbar, L.Carley, Analog Devide-Level Automation, Kluwer Academic Publishers, 1994.
- [2] E.Malavasi, E.Charbon, E.Felt, A.Sangiovanni-Vincentelli, "Automation of IC layout with analog constraints," IEEE Trans. on Computer-Aided Design of IC's and Systems, Vol.15, No.12, pp.1518-1524, 1996.
- [3] H.Murata, K.Fujiyoshi, S.Nakatake and Y.Kajitani, "VLSI Module Placement Based on Rectangle-Packing by the Sequence-Pair," IEEE Trans. on Computer-Aided Design of IC's and Systems, Vol.15 No.12 pp.1518-1524, 1996.
- [4] S.Nakatake, K.Sakanushi, Y.Kajitani and M.Kawakita, "The Channeld-BSG: A Universal Floorplan for Simultaneous Place/Route with IC Applications," Proc. ACM/IEEE Intl. Conf. of CAD, pp.418-425, 1998.
- [5] H.Murata, K.Fujiyoshi and M.Kaneko, "VLSI/PCB Placement with Obstacles Based on Sequence Pair," IEEE Trans. on Computer-Aided-Design of IC's and Systems, Vol.17, No.1, pp.60-68, 1998.
- [6] K.Fujiyoshi and H.Murata, "Arbitrary Convex and Concave Rectilinear Block Packing using Sequence-Pair," Proc. Intl. Symp. on Physical Design, pp.103-110, 1999.
- [7] F.Balasa, K.V.Lampaert, "Module Placement for Analog Layout using the Sequence-Pair Representation," Proc. 36th ACM/IEEE Design Automation Conf., pp.274-279, 1999.