論文 / 著書情報
Article / Book Information

論題(和文)	
Title(English)	Two-layer Bottleneck Channel Track Assignment for Analog VLSI
著者(和文)	谷口 和弥, 田湯 智, 高橋 篤司, モロンゴ マチュー, 南誠, 西岡 克也
Authors(English)	Kazuya Taniguchi, Satoshi Tayu, Atsushi Takahashi, Mathieu Molongo, Makoto Minami, Katsuya Nishioka
出典(和文)	, Vol. 17, , pp. 67-76
Citation(English)	IPSJ Transactions on System LSI Design Methodology, Vol. 17, , pp. 67-76
発行日 / Pub. date	2024, 6
権利情報 / Copyright	

IPSJ Transactions on System LSI Design Methodology Vol.17 67–76 (June 2024)

[DOI: 10.2197/ipsjtsldm.17.67]

Regular Paper

Two-layer Bottleneck Channel Track Assignment for Analog VLSI

Kazuya Taniguchi[1,a)] Satoshi Tayu[1,b)] Atsushi Takahashi[1,c)] Mathieu Molongo[2]
Makoto Minami[2] Katsuya Nishioka[2]

Received: October 4, 2023, Accepted: February 28, 2024

Abstract: Design automation that realizes analog integrated circuits to meet performance specifications in a small area is desired. To reduce the layout area, "Bottleneck Channel Routing" is proposed in which two wires go through a routing track in the bottleneck region. A two-layer routing problem that consists of the bottleneck channel and the adjacent regions where the HV rule is not applicable is defined. The proposed algorithm uses a U-shaped routing model, and generates two-layer routing in which the number of intersections is minimized and the wire of a net includes at most one via. The obtained routing contains no conflicts if the algorithm outputs a feasible solution.

Keywords: channel routing, bottleneck routing, analog VLSI

1. Introduction

Analog VLSI uses a circuit architecture with high tolerance to variation and noise and must meet performance specifications such as current, voltage, phase, cutoff frequency, and signal waveform. The analog layout design is required not to deteriorate the circuit performance. On the other hand, it is important to realize a layout in a small area to reduce manufacturing costs while meeting performance specifications. The objective of our research is to develop a routing framework that enables us to layout a circuit in small area while meeting performance specifications under the assumption that constraints on layout are defined to satisfy performance specifications.

The circuit size of analog VLSI is typically smaller than that of digital VLSI, and analog VLSI often use fewer routing layers than digital VLSI. In VLSI with fewer routing layers, cell-based design where the routing area is defined between cells is often adopted, and may contain bottleneck routing regions which are bottleneck for area reduction. Therefore, the main target of this paper is analog VLSI, and our proposed method contributes to realize a small area layout by reducing the area of bottleneck regions.

In cell-based design [4], the routing area is partitioned into small routing regions called channel or switchbox, and various design flows and algorithms have been proposed. For example, channel routing algorithms were proposed in Refs. [5], [6], [7], [8], [9]. Track assignment procedure considering cross-talk minimization was proposed in Ref. [10]. Sufficient conditions to complete a switchbox routing was discussed in Ref. [11]. Routing architectures which consist of extremal switch-block structures were discussed in Ref. [12]. A design flow without repeating design is discussed in Refs. [13], [14], [15]. For general routing problems, various types of routing algorithms such as maze [16] and A* [17] have been introduced for a single net. For routing of multiple nets, rip-up and reroute technique [18] and length matching [19], [20] to improve the completion ratio of routing and to improve the quality of routing have been discussed. However, they may not fit to cell-based design in which rectangle routing regions without obstacles are defined. Challenges and approaches of VLSI routing was discussed in Ref. [21].

A routing design flow with two-layer HV routing without repeating routing design in which the routing in each layer consists of horizontal segments (H) or vertical segments (V) has been established. However, the obtained layout may contain a routing region which is a bottleneck for area reduction (**Fig. 1** (a)). The height of the bottleneck region may be reduced if two wires share a single routing track, and the layout area may be reduced (Fig. 1 (b)). In such cases, the height of the bottleneck region is up to 50% smaller compared to the region with two-layer HV routing. The reduction of layout area is expected when a bottleneck channel in which two wires share a routing track is introduced at the region that corresponds to a vertex in a critical path in its layout constraint graph that determines the size of the layout area. Of course, there is a trade-off between area and performance, and the circuit performance may be deteriorated due to cross-talk and etc. However, the performance specifications may be met if parallel running length is limited.

In this paper, a bottleneck channel routing framework for cell-based design where HV routing is used for most of regions is discussed. In our routing framework, two wires share a routing track in adjacent layers in the *bottleneck channel*, arbitrary rout-

[1] Tokyo Institute of Technology, Meguro, Tokyo 152–8550, Japan
[2] Jedat, Chuo, Tokyo 104–0043, Japan
[a)] taniguchi@eda.ict.e.titech.ac.jp
[b)] tayu@ict.e.titech.ac.jp
[c)] atsushi@ict.e.titech.ac.jp

The preliminary versions were presented at Refs. [1], [2], [3].

ing is used in *adjacent regions* which is adjacent to the bottleneck channel, and HV routing is assumed in the other regions. Two wires can go through a routing track in adjacent layers only in bottleneck channel, and the situations where the performance specifications are met even if this is allowed are assumed.

In order to give a design guideline for routing framework using bottleneck channel, we define U-shaped two-layer bottleneck channel routing problem and propose algorithm U2TLA-2.0 for this problem. In this problem, routing area consists of a bottleneck channel and adjacent regions. Each net is a two-pin net and has one pin on the upper boundary of each adjacent region. The number of tracks is the half of the number of nets. Algorithm U2TLA-2.0 determines the track and layer assignment of nets for this problem.

According to the track and layer assignment determined by U2TLA-2.0, a routing pattern that consists of three wire segments for each net is obtained. By using U2TLA-2.0, a routing pattern that minimizes the number of intersections when the routing is regarded as planer, and that at most one via is inserted to the wire of each net is obtained. The output of U2TLA-2.0 defines a physical routing pattern that satisfies the connection requirements but may contain conflicts. If U2TLA-2.0 outputs a feasible solution, the physical routing obtained contains no conflict in grid based design. Even if U2TLA-2.0 outputs an infeasible solution, a feasible physical routing may be obtained by modifying the physical routing obtained directly from U2TLA-2.0.

Even though pins are restricted to be placed at the upper boundary of the region in an input of U2TLA-2.0, U2TLA-2.0 will be used as a guideline for general bottleneck channel problems where pins are not restricted at the upper boundary. For example, a physical routing for a general bottleneck channel routing problem can be obtained from a track and layer assignment obtained by U2TLA-2.0 for the U-shaped bottleneck channel routing problem in which the pin order along the boundary is the same as in the general problem.

Also, our bottleneck channel routing framework and algorithm U2TLA-2.0 would be applicable to digital VLSI if it requires a small area with fewer routing layers.

In the following, Sections 2 and 3 introduce the bottleneck channel routing and its subproblem. In Section 4, algorithm U2TLA-2.0 is proposed. In Sections 5 and 6, discussions and experiments related to U2TLA-2.0 are given, respectively, and the conclusion is stated in Section 7.

2. Bottleneck Channel Routing

A routing problem is to find a better routing pattern that satisfies the connection requirement under the design rule. The connection requirement among pins is called a *net*. Pins of all nets must be connected by wire that can pass several routing layers. *Vias* must be inserted to a wire when the routing layer of the wire is changed. If wires of different nets intersect in the planar projection of a routing pattern, they must be assigned to different layers at the intersection. In grid based design, the wires of different nets have a *conflict* if they share the same coordinate in the same layer. A routing pattern that satisfies the connection requirement in grid-based design is *infeasible* if there is a conflict, *feasible* otherwise.

Bottleneck channel routing problem is defined on routing area that consists of a bottleneck channel and adjacent regions on both sides as shown in **Fig. 2**. Pins of each net are placed on the boundary of adjacent regions. Pins are virtual and are assumed to be accessible from any layer. A wire which connects pins of a net goes through a track in the bottleneck channel, and two wires can go through a track in adjacent layers.

In order to give a design guideline for routing framework using bottleneck channel, U-shaped two-layer bottleneck channel routing problem is defined as follows: the routing area consists of a bottleneck channel and adjacent regions; each net is a two-pin net and has one pin on the upper boundary of each adjacent region; the number of tracks is the half of the number of nets.

The problem instances for this problem is limited, but it would

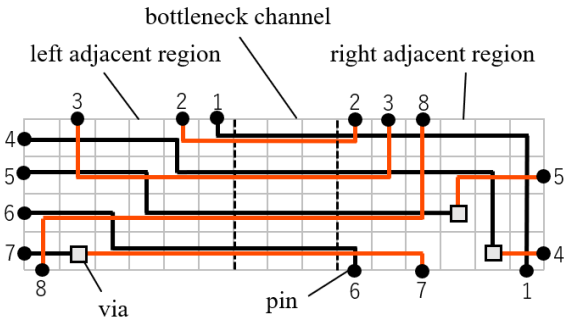
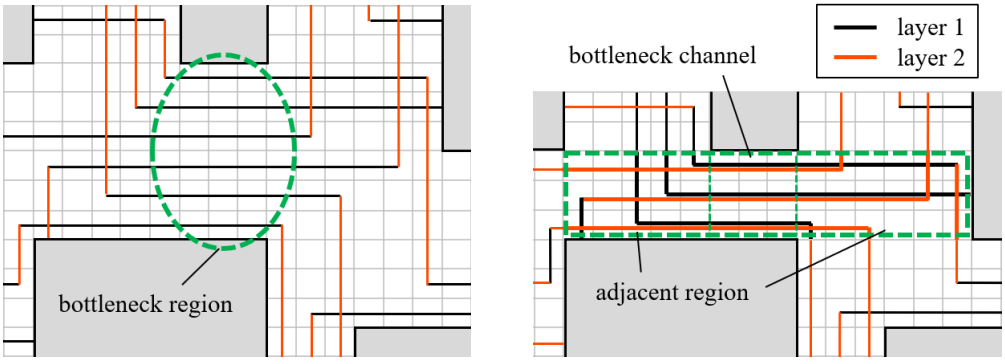


Fig. 2 Bottleneck channel routing.



(a) HV Routing (b) Bottleneck Channel Routing

Fig. 1 Circuit layouts by the same global placement and routing.

be used as a guideline to obtain a routing pattern for a general bottleneck channel routing problem. For example, a multi-pin net is able to assume to be decomposed into two-pin nets appropriately in advance so that the congestion at the bottleneck channel is reduced. In case that the number of tracks available at the bottleneck channel is larger than the half of the number of nets, dummy nets can be inserted to fit the problem formulation. Once a physical routing pattern in this problem formulation is obtained, post processing would be applied to fit an actual situation, and to improve the performance, but it would be in the future works.

3. U-shaped Two-layer Bottleneck Channel Routing Problem

In this section, U-shaped two-layer bottleneck channel routing problem shown in **Fig. 3** in which wires of each net are "U-shaped" and the number of tracks is the half of the number of nets is formulated.

In a U-shaped two-layer bottleneck channel routing problem, routing area G_m for the set N of $2m$ two-pin nets is modeled by the routing grid ($-2m \le x \le 2m,\ 0 \le y \le m$) shown in **Fig. 4** where the y-axis corresponds to the degenerated *bottleneck channel*, the region $x < 0$ corresponds to the *left-adjacent region*, and the region $x > 0$ corresponds to the *right-adjacent region*. A pin is placed at grid point $(x, 0)$ on the x-axis where x is an integer ($1 \le |x| \le 2m$). A pin placed on grid point $(x, 0)$ is called left pin if $-2m \le x \le -1$, and right pin if $1 \le x \le 2m$. Each two-pin net consists of one left pin and one right pin. Let $l(n)$ and $r(n)$ be the x-coordinate of the left and right pins of net n, respectively. Pin position is specified by the left and right pin sequences $P_{\text{left}} = (l_1, l_2, \ldots, l_{2m})$ and $P_{\text{right}} = (r_1, r_2, \ldots, r_{2m})$ where $l_i = n$ if $l(n) = -i$, and $r_i = n$ if $r(n) = i$ ($\forall n \in N$). In the following, it is assumed that $n_i = i$ and $l(n_i) = -i$ for clarity. Track t ($1 \le t \le m$) is the grid line connecting $(-2m, t)$ and $(2m, t)$.

The output of this problem are the track assignment of nets A_T and the layer assignment of nets A_L.

U-shaped two-layer bottleneck channel routing problem
Input:
- net set $N = \{n_1, n_2, \ldots, n_{2m}\}$ $(m > 0)$
  - $n = (l(n), r(n))$ where $-l(n), r(n) \in \{1, 2, \ldots, 2m\}$ $(\forall n \in N)$
  - $l(n_i) \neq l(n_j)$ and $r(n_i) \neq r(n_j)$ if $i \neq j$

Output:
- track assignment of nets
  $A_T : N \to \{1, 2, \ldots, m\}$
- layer assignment of segments of nets
  $A_L : N \to \{1, 2\} \times \{1, 2\} \times \{1, 2\}$
  (left vertical, horizontal, right vertical)

Fig. 3 Problem formulation.

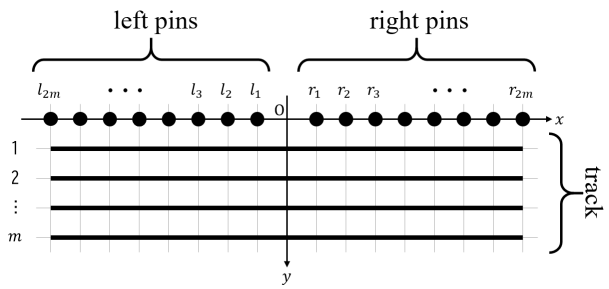


Fig. 4 Routing area G_m, tracks, and pins.

In a U-shaped two-layer bottleneck channel routing problem, an output has to satisfy the following three conditions.

- The wire of each net consists of one horizontal and two vertical segments.
- Each segment is assigned to either layer 1 or layer 2.
- At least one vertical segment of each net is assigned to the same layer where the horizontal segment of the net is assigned.

Each track is assigned two nets one on layer 1 and the other on layer 2. The wire of a net consists of two vertical segments whose x-coordinates are the x-coordinates of pins of the net and one horizontal segment which is assigned to a track.

In a routing pattern satisfying the conditions above, a routing patten in which at most one via is inserted for each net is obtained. If two vertical segments of a net are assigned to the layer of the horizontal segment of the net, then no via is inserted in the wire of the net, and if one is assigned to a different layer, then one via is inserted between the vertical segment and the horizontal segment.

Output of U2TLA-2.0 is the track and layer assignment. The routing pattern is uniquely determined under the three conditions above if the track and layer assignment is determined. Among routing patterns which satisfy the conditions above, U2TLA-2.0 uses one of four routing patterns for each net shown in **Fig. 5**. U2TLA-2.0 prefers to use layer 1 and layer 2 for left and right vertical segments, respectively. Another layer is used if the number of vias can be reduced without creating conflicts.

Note that a pair of nets assigned to the same track have a conflict if a vertical segment, which is assigned to the different layer of the horizontal segment of the net, is located within the interval of the other net, then the vertical segment and the horizontal segment of the other net have conflict as shown in **Fig. 6**, and the routing pattern is infeasible. Otherwise, the pair of nets have no conflict. U2TLA-2.0 avoids creating this type of conflicts as much as possible.

In order to explain the properties of problem and algorithm, the followings are defined in terms of input and track assignment of nets (see **Fig. 7**). Note that algorithm U2TLA-2.0 determines

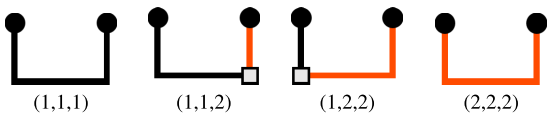


Fig. 5 Routing pattern of a net used by U2TLA-2.0.

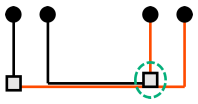


Fig. 6 Routing pattern of a pair of nets with a conflict.

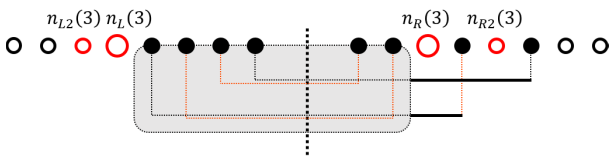


Fig. 7 Variables and functions used in U2TLA-2.0 and the fixed region when track-2 assignment is finished (Pins of a net are drawn in black if the track and layer assignment of the net is determined, white otherwise).

track assignment from the upper track.

- $N(t) := \{n \mid A_T(n) \not\leq t-1\}$ (the set of nets not assigned to track from 1 to $t-1$)
- $n_L(t)$: net n s.t. $l(n)$ is maximum in $N(t)$
- $n_{L2}(t)$: net n s.t. $l(n)$ is maximum in $N(t) \setminus \{n_L(t)\}$
- $n_R(t)$: net n s.t. $r(n)$ is minimum in $N(t)$
- $n_{R2}(t)$: net n s.t. $r(n)$ is minimum in $N(t) \setminus \{n_R(t)\}$

If there are various track and layer assignments that satisfies the conditions, an assignment that can enable us to derive a feasible routing is preferred. Also, the following indices defined in a derived feasible routing will be used as evaluation of solutions.

(1) the number of vias that are not at the end of segment.
(2) the number of vias that are not at grid points.
(3) the number of vias.

4. Algorithm U2TLA-2.0

4.1 Overview

The overview of our proposed algorithm U2TLA-2.0 for U-shaped two-layer routing problem is given here. U2TLA-2.0 consists of main routine shown in **Fig. 8** and two subroutines shown in **Fig. 9**.

U2TLA-2.0 first partitions nets into clusters by NetCluster, and executes one or two threads to determine the track and layer assignment for each cluster by U2TLA. The track and layer assignment for each cluster is independent of other cluster, and the better track and layer assignment is selected if two threads are executed for a cluster. In each thread, U2TLA iteratively assigns the horizontal segment of a pair of nets from the upper track so that invariant condition defined below is satisfied. When the pair of nets are assigned to a track, the *fixed region* shown as a gray region in Fig. 7 which is not used in latter assignment is defined, although a wire segment of an assigned net may exist outside the fixed region.

The *invariant condition* is that any horizontal segments of assigned nets assigned to layers 1 and 2 do not exist in the left and the right of fixed region, respectively. U2TLA-2.0 classifies situations when a pair of nets are assigned to a track into eight cases under the assumption that the invariant condition is satisfied, and it determines the track and layer assignment to keep the invariant condition as shown in **Fig. 10**. Among the eight situations shown in Fig. 10, only case (B)-11 derives an infeasible routing pattern. U2TLA-2.0 runs two thread when case (B)-00 occurs to avoid (B)-11 as much as possible.

U2TLA-2.0
Input: N
Output: $A_T(N)$, $A_L(N)$
$(N_1, N_2, \ldots, N_c)$:= **NetCluster**(N)
$t_0 := 0$
for$(1 \leq i \leq c)$
  ***** run assignment thread *****
  $(A_T(N_i), A_L(N_i), f)$:= **U2TLA**$(N_i, t_0, 1)$
  ***** if (B)-00 $(n_L(t_0) = n_R(t_0))$ then run second thread *****
  if(f = **FALSE** && $n_L(t_0) = n_R(t_0)$)
    $(A_T(N_i), A_L(N_i), f)$:= **U2TLA**$(N_i, t_0, 2)$
  endif
  $t_0 := t_0 + |N_i|/2$
endfor

Fig. 8 Algorithm U2TLA-2.0 main.

In Fig. 10, the left and right figures show before track assignment of nets L, R and the results of track assignment of L, R, respectively. In the left figures, the name of nets $n_L(t), n_{L2}(t), n_R(t)$, and $n_{R2}(t)$ are surrounded by frames, and the name of nets $n_L(t)$ and $n_R(t)$ are shaded. The symbol "#" represents the nets either L or R. The arrows extend from the origin to L or R, respectively. The names of unassigned nets whose pins are placed inside of fixed region or outside of pins surrounded by frames are omitted.

4.2 Track and Layer Assignment

The track and layer assignment for each cluster by U2TLA is explained here.

The fixed region which is not used in latter assignment is defined during execution. It contains only pins and wires of assigned nets. The fixed region when the pair of nets assigned to

NetCluster
Input: N ($|N| = 2m$)
Output: $(N_1, N_2, \ldots, N_c)$
$t := 1$
$c := 0$, $N_c := \emptyset$
$N' := N$
while$(t \leq m)$
  L := net n s.t. $l(n)$ is maximum in N'
  R := net n s.t. $r(n)$ is minimum in N'
  if$(l(L) = -2t+1$ && $r(R) = 2t-1)$
    $c := c+1$, $N_c := \emptyset$
  endif
  if$(L = R)$ $R := n_{R2}(t)$
  $N_c := N_c \cup \{L, R\}$
  $N' := N' \setminus \{L, R\}$
  $t := t+1$
endwhile

U2TLA
Input: N_i, t_0, s
Output: $A_T(N_i)$, $A_L(N_i)$, f
Step 1. (initialization)
  $t := t_0$
  $p_{\text{left}}(t) := 0$, $p_{\text{right}} := 0$
  f := **TRUE**
Step 2. (assignment thread)
  while$(t \leq t_0 + |N_i|/2)$
    Step 2.1. (selection of L, R)
      $L := n_L(t)$, $R := n_R(t)$
      if$(L = R)$
        switch$(l(L) > -2t+1, r(R) < 2t-1)$
          case$(0,0)$:
            if$(s = 1)$ $R := n_{R2}(t)$
            if$(s = 2)$ $L := n_{L2}(t)$
          case$(0,1)$:
            $L := n_{L2}(t)$
          case$(1,0)$:
            $R := n_{R2}(t)$
          case$(1,1)$:
            $R := n_{R2}(t)$
            f := **FALSE**
        endswitch
      endif
    Step 2.2. (assignment of L, R)
      $A_T(L) := t$, $A_T(R) := t$
      $A_L(L) := (1,1,1)$, $A_L(R) := (2,2,2)$
      if$(r(L) < p_{\text{right}}(t))$ $A_L(L) := (1,1,2)$
      if$(p_{\text{left}}(t) < l(R))$ $A_L(R) := (1,2,2)$
    Step 2.3. (update)
      $p_{\text{right}}(t+1) := \max\{p_{\text{right}}(t), r(L)\}$
      $p_{\text{left}}(t+1) := \min\{p_{\text{left}}(t), l(R)\}$
      $t := t+1$
  endwhile

Fig. 9 Algorithm U2TLA-2.0 subroutines.

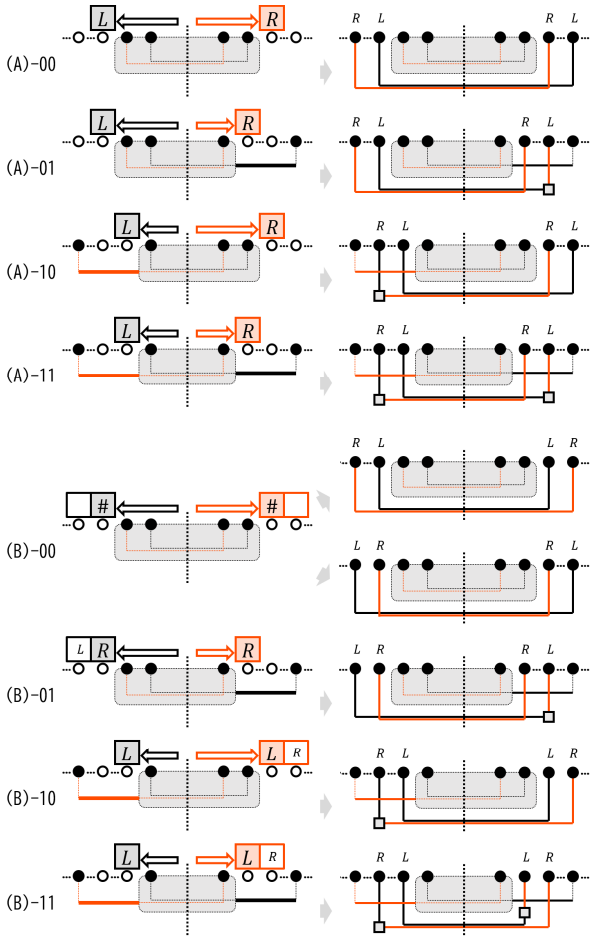


Fig. 10 Eight situations of routing region ($L, R, \#$: nets assigned to the next track, gray: fixed region).

track t is the region whose x-coordinate spans $\max\{l(n) \mid n \in N(t)\} + 1 \le x \le \min\{r(n) \mid n \in N(t)\} - 1$ and whose y-coordinate spans $1 \le y \le t - 1$ (see Fig. 7). Note that $N(t)$ is the set of unassigned nets, and $\max\{l(n) \mid n \in N(t)\}$ is the x-coordinate of the pin which is closest to the origin among left pins of unassigned nets.

The situation of routing region that satisfies the invariant condition is classified into eight cases (Fig. 10). The eight cases are defined by the combination of whether (A) $n_L(t) \neq n_R(t)$ or (B) $n_L(t) = n_R(t)$, and whether the horizontal segments extend beyond the fixed region to the left or right (00, 01, 10, 11). Note that $n_L(t)$ and $n_R(t)$ is the net whose left and right pins are closest to the origin among unassigned nets, respectively.

U2TLA selects a pair of nets (L, R) so that pins of L and R are the inner most left and right pins of unassigned net, respectively. In case that the inner most left and right pins are different, L and R are uniquely determined. That is, $(L, R) = (n_L(t), n_R(t))$ if $n_L(t) \neq n_R(t)$ (Fig. 10 (A)). In case that they are the same, one of second inner pin is selected. That is, either $(L, R) = (n_{L2}(t), n_R(t))$ or $(L, R) = (n_L(t), n_{R2}(t))$ if $n_L(t) = n_R(t)$ (Fig. 10 (B)). The former is used in (B)-01, and the latter is used in (B)-10 and (B)-11. In case that (B)-00, it is determined by U2TLA-2.0.

U2TLA assigns the horizontal segments of L and R to layers 1 and 2, respectively, and the invariant condition is satisfied at the next track. The left vertical segment of L and the right vertical segment of R are assigned to layer 1 and layer 2, respectively. While, the layer assignment of the right vertical segment of L and the left vertical segment of R depend on the situation. If the right vertical segment of L is assigned to layer 2, then no conflict occurs because of the invariant condition. If no conflict occurs when it is assigned to layer 1, then it is assigned to layer 1 to reduce the number of vias. Similarly, the assignment of the left vertical segment of R is determined.

4.3 Cluster Extraction

A cluster is extracted from the input net set by subroutine NetCluster. A cluster is defined as a minimal consecutive subsequence $(P'_{\text{left}}, P'_{\text{right}})$ of the input pin sequence that consists of pins of nets of even number. That is, $P'_{\text{left}} = (l_{2i+1}, \ldots, l_{2j})$, $P'_{\text{right}} = (r_{2i+1}, \ldots, r_{2j})$ and $n \in P'_{\text{left}}$ if and only if $n \in P'_{\text{right}}$. There is a topological routing that satisfies the connection requirement where wires of nets do not intersect if they belong to different clusters. In U-shaped routing, the track and layer assignment for each cluster can be determined independent of other cluster, and U2TLA-2.0 determines the track and layer assignment of clusters from inner to outer so that wires of nets do not intersect if they belong to different clusters.

Subroutine NetCluster extracts clusters by detecting the borders of clusters in pin sequence. NetCluster forms a cluster by adding two nets whose pins are inner most among remaining nets iteratively. If the coordinates of pins of adding nets are $-2t + 1$ and $2t - 1$ at t-the addition, then the pins of all the nets added before locate inner side of the adding nets. This corresponds to (A)-00 or (B)-00, and means the border of a cluster. NetCluster forms a new cluster, and adds two nets to the new cluster.

4.4 Assignment Thread

Among the eight situations shown in Fig. 10, only case (B)-11 derives an infeasible assignment. Depending on the net selection so far, it may be possible to avoid the occurrence of (B)-11.

There is no choice in net pair selection for cases (A) to keep the invariant condition. For cases (B), $n_L(t)$ $(= n_R(t))$ is selected as one of a pair, and there may exist a choice in the selection of the other net.

In cases (B)-01 and (B)-10, the layer of horizontal segment of $n_L(t)$ $(= n_R(t))$ is determined to avoid sharing the same coordinate on the same layer at the track, and the other net to keep the invariant condition is unique. For case (B)-11, feasible assignment cannot be obtained, and U2TLA-2.0 chooses the other net whose right pin is the next inner-most.

In case (B)-00, the assignment of the net pair causes no conflict at the track. However, depending on the net pair selection, case (B)-11 occurs later.

In **Figs. 11** and **12**, the same routing problem is given, and case (B)-00 occurs on track-2 where $n_L(t) = n_R(t) = n_3$. In Fig. 11, the horizontal segment of n_3 is assigned to layer 1, and n_8 is selected as R to assign track-2 and a feasible routing pattern is obtained. In Fig. 12, the horizontal segment of n_3 is assigned to layer 2, and n_4 is selected as L to assign track-2 and a conflict occurs on track-4.

In case (B)-00, U2TLA-2.0 tries two assignment threads in which (L, R) is either $(n_L(t), n_{R2}(t))$ or $(n_{L2}(t), n_R(t))$ respectively,

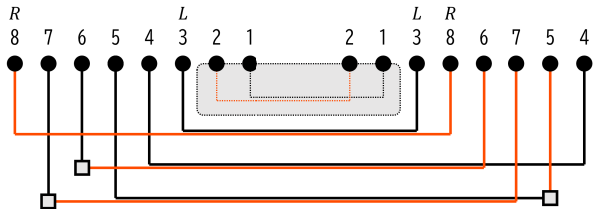


Fig. 11 Routing pattern obtained by U2TLA-2.0.

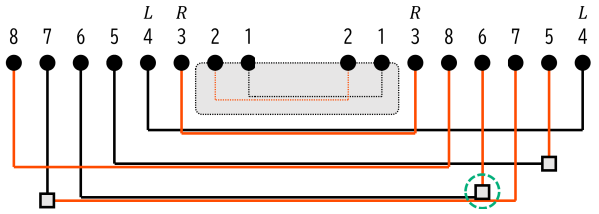


Fig. 12 Infeasible routing pattern.

Table 1 The behavior of U2TLA-2.0.

t	L	R	$l(R)$	$p_{\text{left}}(t)$	$r(L)$	$p_{\text{right}}(t)$	Situation
1	n_1	n_2	-2	0	2	0	(A)-00
2	n_3	n_8	-8	-2	3	2	(B)-00
3	n_4	n_6	-6	-8*	8	3	(A)-10
4	n_5	n_7	-7	-8*	7	8*	(A)-11

and selects one of them when two threads end. In case that one thread is feasible and the other is infeasible, U2TLA-2.0 selects one which is feasible. In case that both threads are feasible or infeasible, U2TLA-2.0 assign the layer of the horizontal segment of $n_L(t)$ ($= n_R(t)$) to layer 1 or layer 2, respectively.

From the definition of $n_L(t)$ and $n_R(t)$, $l(n_L(t)) \geq -2t + 1$ and $r(n_R(t)) \leq 2t - 1$ hold. Cases (A)-00 and (B)-00 occur when pins of nets not belonging to $N(t)$ are all placed between $l(n_L(t))$ and $r(n_R(t))$. In other words, the condition for cases (A)-00 and (B)-00 on track t is that $l(n_L(t)) = -2t + 1$ and $r(n_R(t)) = 2t - 1$.

4.5 Example

The behavior of U2TLA-2.0 is shown by using the problem given in Fig. 11 where $P_{\text{left}} = (n_1, n_2, n_3, n_4, n_5, n_6, n_7, n_8)$ and $P_{\text{right}} = (n_2, n_1, n_3, n_8, n_6, n_7, n_5, n_4)$. The value of each variable at t is shown in **Table 1**. Column "Situation" in Table 1 corresponds to the situation of routing region in Fig. 10. A routing pattern obtained by U2TLA-2.0 is shown in Fig. 11.

A via is required to insert to the wire of R if $p_{\text{left}}(t) < l(R)$ in Step 2.2 of U2TLA. Similarly, a via is required to insert to the wire of L if $r(L) < p_{\text{right}}(t)$. Note that $p_{\text{left}}(t)$ and $p_{\text{right}}(t)$ are the minimum and maximum x-coordinates of the left and right pins of nets which are assigned to a track either from 1 to $t - 1$ as shown in Fig. 7, respectively. In Table 1, the symbol "*" after the value of $p_{\text{left}}(t)$ ($p_{\text{right}}(t)$) represents that a via is required to insert to the wire of R (L). If via is required to insert to R (L), p_{left} (p_{right}) remains the same value in Step 2.3 of U2TLA. Otherwise, it is updated by substituting the second term in Step 2.3.

5. Discussion

5.1 Validity of U2TLA-2.0

In this section, the validity of U2TLA-2.0 is discussed. It is shown that in the routing pattern obtained by U2TLA-2.0 is feasible if the situations of routing region of all tracks are other than (B)-11.

It is obvious from the definition of segments of a net that the segments of each net form a wire connecting both pins of the net. It is shown that the following two propositions are satisfied in the track assignment for track t.

- The wire of a net assigned to track t has no conflicts with wires of nets assigned before track t.
- The wires of nets assigned to track t have no conflicts with each other if the situation of routing region of t is other than (B)-11.

Note that tracks do not intersect with each other, and that a vertical grid line is exclusively used by a net.

Lemma 1. *The wire of a net assigned to track t has no conflicts with wires of nets assigned before track t.*

Proof. Each segment of net L selected in Step 2.1 of U2TLA has no conflicts with wires of nets assigned before track t is shown.

(a) left vertical segment
The left vertical segment of L is assigned to layer 1 in Step 2.2, and only may has conflicts with horizontal segments in layer 1. However, the horizontal segment of a net assigned before track t of layer 1 terminates to the right of the left vertical segment of L. If the horizontal segment of net n assigned to the layer 1 does not terminate, then it contradicts the net selection in Step 2.1 since L should be selected before n. Therefore, the left vertical segment of L has no conflicts with wires of nets assigned before track t.

(b) horizontal segment
Since the wire of a net assigned before track t exists in the region $y \leq t - 1$, they have no conflicts with the horizontal segment of L on $y = t$.

(c) right vertical segment
The right vertical segment of L is assigned to layer 2 if the segment intersects with a horizontal segment assigned to layer 1, and is assigned to layer 1 otherwise. The horizontal segment of a net assigned before track t of layer 2 terminates to the left of the right vertical segment of L because of the net selection in Step 2.1, and have no conflicts with the right vertical segment of L.

From (a), (b) and (c), it is confirmed that the segments of the net L have no conflicts with wires of nets assigned before track t. Similarly, it can be confirmed that the segments of R selected in Step 2.1 have no conflicts with wires of nets assigned before track t, but the description is omitted here. □

Lemma 2. *The wires of nets assigned to track t have no conflicts with each other if the situation of routing region of t is other than (B)-11.*

Proof. The wires of L and R assigned to track t have no conflicts with each other is shown. The segments that share coordinates among the wires of L and R are either (a) two horizontal segments or (b) one horizontal and one vertical segments.

(a) The horizontal segments of L and R have no conflicts with each other because they are assigned to different layers.

(b1) The horizontal segment of L (R) has no conflicts with the right (left) vertical segment of R (L) because they are assigned to different layers from Step 2.2.

(b2) The conflict between the horizontal segment of L (R) with

the left (right) vertical segment of R (L) is checked in the following. In case that $n_L(t) \neq n_R(t)$, they have no conflicts because they do not share coordinates. The horizontal segment and the vertical segment share coordinates only if $n_L(t) = n_R(t)$ (see Fig. 10 (B)). If either $l(n_L(t)) = -2t + 1$ or $r(n_R(t)) = 2t - 1$ (Fig. 10 (B)-00, 01, 10), then they are assigned to different layers, and they have no conflicts with each other. Otherwise, they conflict with each other, and the situation of routing region of t is (B)-11 (Fig. 10 (B)-11).

Therefore, the wires of L and R assigned to track t have no conflicts with each other. □

Note that a conflict occurred with (B)-11 can be resolved by inserting the via to the middle of the vertical segment between track $t-1$ and track t.

Theorem 1. *A routing which satisfies the connection requirements without conflicts is obtained by U2TLA-2.0 in case that the situations of routing region of all tracks are other than (B)-11.*

Proof. It is obvious from Lemmas 1 and 2 that the net assigned to track t can be realized without conflicts with wires of nets assigned before track t if the situations of routing region of all tracks are other than (B)-11. □

The time complexity of U2TLA-2.0 is $O(m)$ where m is the number of tracks, since each step in U2TLA takes a constant time, and the number of threads executed for each net is at most two since the track and layer assignment for each cluster is independent of other cluster.

5.2 A Sufficient Condition for Feasible Solution

The necessary and sufficient condition on the input of a U-shaped two-layer routing problem that have a feasible solution is not known. The following theorem gives a sufficient condition on inputs that have a feasible solution.

Theorem 2. *In the U-shaped two-layer bottleneck channel routing problem with net set N, a feasible solution exists if there are no nets $a, b, c, u \in N$ such that $l(a) < l(u) < l(b), r(a) < r(u) < r(b), l(c) < l(u), r(u) < r(c)$.*

Proof. It is enough to show that there are nets satisfying the condition when U2TLA-2.0 outputs an infeasible solution. U2TLA outputs an infeasible solution only if the situation of routing region is (B)-11 for some track t. Suppose that $n_L(t) = n_R(t)$, $l(n_L(t)) > -2t + 1$, and $r(n_R(t)) < 2t - 1$ are satisfied for track t. Let net u be L ($= n_L(t) = n_R(t)$).

The left vertical segment of u intersects the horizontal segments assigned to layer 2 of the track upper than the horizontal segment of u. There exists a net a satisfying $l(a) < l(u)$, and from Step 2.1, $r(a) < r(u)$.

The right vertical segment of u intersects the horizontal segments assigned to layer 1 of the track upper than the horizontal segment of u. There exists a net b satisfying $r(u) < r(b)$, and from U2TLA, $l(u) < l(b)$.

The left and right pins of a net whose horizontal segment is assigned to the same track as of u are placed outside of the left and right pins of u. There exists a net c satisfying $l(c) < l(u)$, $r(u) < r(c)$.

The existence of nets that satisfies the condition is shown and Theorem 2 is proved. □

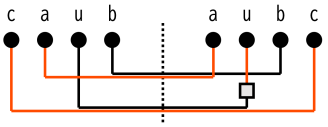


Fig. 13 Infeasible assignment.

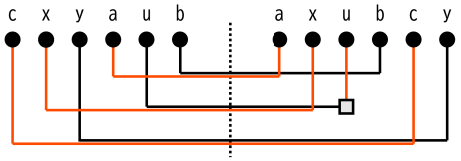


Fig. 14 The inverse of Theorem 2 does not hold.

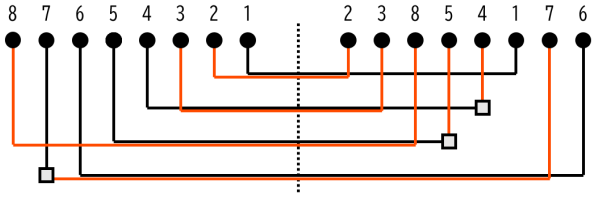


Fig. 15 U-shaped routing corresponds to the routing in Fig. 2.

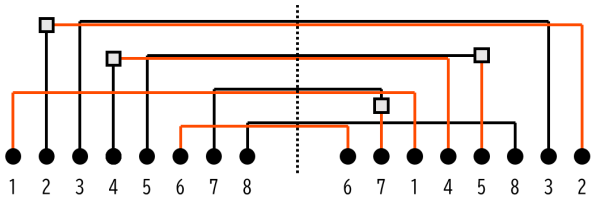


Fig. 16 Routing for the same pin sequence along boundary in Fig. 15.

Figure 13 shows an example of an infeasible solution. In case that the number of nets is four, there are four patterns of pin sequences in which the outputs are infeasible. Even in case that the net set contains four nets violating the prerequisites of Theorem 2, a feasible physical routing may exist as shown in **Fig. 14**.

5.3 Ideas for Extensions

A track assignment and a layer assignment of the topological routing obtained by U2TLA-2.0 could be utilized to find the physical routing of a bottleneck channel problem. Even if an infeasible solution is obtained in U-shaped routing problem, the conflict may be resolved in the physical routing in corresponding bottleneck channel problem.

An example of a method to obtain a physical routing of a general bottleneck channel problem using U2TLA-2.0 is given here. First, the U-shaped two-layer routing problem is defined from a bottleneck channel problem so that both of them have the same pin sequence along the boundary, and the track and layer assignment is determined by U2TLA-2.0. Next, the topology of the routing derived by the track and layer assignment is extracted. Finally, the physical routing of the bottleneck channel problem is determined according to the obtained topology of the routing. A method to convert topological routing to physical routing is in our future works. The U-shaped routing problem shown in **Fig. 15** corresponds to the bottleneck routing problem shown in Fig. 2. Note that the pin sequence along the boundary is the same between them.

Different topological routing may affect the feasibility of the

corresponding physical routing. Other topological routing might be obtained if other U-shaped two-layer routing problem is defined. For example, the routing problem shown in **Fig. 16** is obtained in which the pins are placed on the lower boundary without changing their order from the problem shown in Fig. 2. The routing pattern shown in Fig. 16 is obtained by U2TLA-2.0 by assigning track from below with different net selection order. Two routing patterns obtained may have different track assignments and different the number of vias. It is also possible that one is feasible and the other is infeasible. This flexibility will help us to have more options when considering physical routing of a bottleneck channel problem.

In U-shaped two-layer routing problem, the number of tracks is set to the half of the number of nets. Conflicts can be resolved if more tracks are available. A conflict occurred at a track in (B)-11 can be resolved if one more additional track is available. For example, in Fig. 12, the horizontal segment of n_7 has conflict with the right vertical segment of n_6, and the conflict can be resolved by using one additional track, in which n_7 is assigned to track-5.

Conflicts can be resolved if pin sequence can be modified. Let's consider the case that a conflict occurs with (B)-11 on track t. The conflict occurs between the right vertical segment of $n_L(t)$ and the horizontal segment of $n_{R2}(t)$. The right pin of $n_L(t)(= n_R(t))$ and the right pin of $n_{R2}(t)$ are adjacent, and the conflict can be resolved by swapping the location of right pins of these nets. For example, in Fig. 12, the conflict can be resolved by swapping the location of right pins of n_6 and n_7.

6. Experimental Results

U2TLA-2.0 is implemented in Python 3.8.10, and executed on a PC with 3.30GHz Intel Core i7 CPU. They are applied to two types of problem instances with different properties. For both types, one hundred instances are generated for each net size where the left pin sequence P_{left} is fixed as $(n_1, n_2, \ldots, n_{2m})$.

In type one, instances are generated randomly for net size 8, 32, 128, and 512. The right pin sequence P_{right} is generated randomly.

In type two, instances assuming bus type routing are generated. In type two, an instance is obtained by concatenating randomly generated 8-net instances. The net size 32 (8×4), 128 (8×16), and 512 (8×64) are generated.

Table 2 The average number of tracks used to obtain a feasible solution.

#nets	HV		bottleneck	
8	8	(1.00)	4.63	(0.58)
32	32	(1.00)	17.55	(0.55)
128	128	(1.00)	66.47	(0.52)
512	512	(1.00)	258.93	(0.51)

In **Table 2**, the comparison between HV-routing and bottleneck routing by using type one instances is shown. In the table, the average numbers of tracks used to obtain a routing pattern without conflicts are shown. In HV routing, the number of tracks used is equals to the number of nets. A feasible routing pattern is obtained by left-edge algorithm [5] since there is no vertical constraint among the horizontal segments of nets. In bottleneck routing, the track assignment is determined by U2TLA-2.0. In cases that the track assignment contains conflicts, additional tracks are inserted to resolve these conflicts. The number of tracks inserted is equals to the number of conflicts in the track assignment by U2TLA-2.0. The average number of tracks used in bottleneck channel is up to 50% of the number of tracks used in HV-routing. It is confirmed that the ratio approaches 50% when the number of nets increases.

In **Table 3**, the details of experimental results of U2TLA-2.0 for instances of both types are shown. "#via", "ratio", "probability", "#conf." represent the average number of vias, the ratio of instances where the output contains no conflict, the probability of tracks that contain conflicts, and the number of tracks that contain conflicts in the output, respectively. "one" and "two" represent the average number of one-thread-clusters and two-thread-clusters that corresponds to (A)-00 and (B)-00, respectively. "#avoid" represents the number of conflicts that are not included in the output since the assignments by the other thread are adopted.

If the output contains even one conflict, the solution is infeasible. However, if the number of nets increase, the probability of tracks that contains a conflict reduces. For instances of 512 nets, the number of tracks that contains a conflict is at most seven and is less than three in average. In case that instances are generated intentionally to increase the number of clusters, the average of the probability is high since the instances are generated by concatenating small instances.

The number of conflicts is equal to the number of tracks where case (B)-11 occurred. Note that a feasible assignment is obtained when the number of conflicts is zero. The number of clusters in which one assignment thread is run is equal to the number of tracks where case (A)-00 occurred. The number of clusters in which two assignment threads are run is equal to the number of tracks where case (B)-00 occurred. The rightmost column in Table 3 shows the average number of clusters in which two assignment threads are run, and one of which has conflicts and the other does not have.

As confirmed in Table 3, the number of clusters is one in most cases when instances are generated randomly. In case that instances are generated intentionally to increase the number of clus-

Table 3 Experimental results of U2TLA-2.0 for each net size.

#nets	time [μs]	#via	ratio	probability	#conf.			one		two		
	ave	ave	[%]	[%]	ave	max	min	#	#conf.	#	#conf.	#avoid
8	25.530	2.95	50	15.75	0.63	2	0	0.95	0.59	0.18	0.04	0.05
32	88.463	24.16	18	9.75	1.55	4	0	0.98	1.52	0.02	0.03	0
128	334.974	116.98	7	3.86	2.47	6	0	1.00	2.47	0	0	0
512	2837.368	498.30	6	1.14	2.93	7	0	1.00	2.93	0	0	0
32 (8×4)	80.728	12.25	3	14.81	2.37	5	0	3.82	2.29	0.65	0.08	0.12
128 (8×16)	219.531	49.13	0	14.80	9.44	16	3	15.00	9.07	2.49	0.37	0.51
512 (8×64)	643.619	196.84	0	14.44	36.77	50	21	59.71	35.08	10.09	1.69	2.28

ters, the conflict probability of tracks remains high, and the feasible instance ratio rapidly decreases to 0. Also, it is confirmed that the ratio of two-thread-clusters is small and that the number of conflicts removed by the selection of thread is limited. However, in practice, it seems that both type of instances are unlikely to be used. Instances in which no conflicts occur will be used according to the analysis.

Even though an output of U-shaped two-layer bottleneck channel routing problem contains conflicts, these conflicts may be resolved if the tight routing conditions are relaxed. Also, it is easy to resolve the conflicts if few more tracks are available. A physical routing of a general bottleneck channel routing problem derived from topological routing obtained by U2TLA-2.0 may not contain conflicts. The analysis of U2TLA-2.0 behavior will be utilized in global routing phase to generate a bottleneck channel routing problem in which no conflicts occur.

7. Conclusion

In this paper, bottleneck channel routing that enable us to reduce the routing area is proposed. For the two-layer bottleneck channel problem with 2-pin net, algorithm U2TLA-2.0 which obtains a routing in which the wire of a net uses at most one via is proposed. A feasible physical routing in grid based design is obtained according to the track and layer assignment by U2TLA-2.0 if all pins are placed on the upper boundary of the routing area and if U2TLA-2.0 defines the output except case (B)-11. Even if U2TLA-2.0 outputs an infeasible solution, a corresponding two-layer topological routing that satisfies the connection requirement is obtained, and a feasible physical routing may be obtained if vias can be inserted in the middle of a segment, or if more tracks are available, or if the pin sequence can be modified.

Our future works include an extension to the multi-pin net problem, an extension to three or more routing layers, and the development of an algorithm that obtains a physical routing of general bottleneck channel routing utilizing the topological routing obtained by U2TLA-2.0. The routing shown in Fig. 2 is obtained from the topological routing obtained by U2TLA-2.0 for the U-shaped physical routing problem shown in Fig. 15. The routing pattern defined according to the output of U2TLA-2.0 would be utilized as an initial solution, and a better physical routing would be explored by applying post processing.

References

[1] Taniguchi, K., Tayu, S., Takahashi, A., Todoroki, Y. and Minami, M.: Bottleneck Channel Routing to Reduce the Area of Analog VLSI, *IEICE Tech. Rep.*, Vol.121, No.412, pp.7–12 (2022). (in Japanese).

[2] Taniguchi, K., Tayu, S., Takahashi, A., Todoroki, Y. and Minami, M.: Bottleneck Channel Routing to Reduce the Area of Analog VLSI, *Proc. SASIMI 2022*, pp.26–31 (2022).

[3] Taniguchi, K., Tayu, S., Takahashi, A., Molongo, M., Minami, M. and Nishioka, K.: Track Assignment considering Routing Crossing Relations to Improve Feasibility in Bottleneck Channel Routing, *IEICE Tech. Rep.*, Vol.122, No.402, pp.149–154 (2023). (in Japanese).

[4] Kahng, A., Lienig, J., Markov, I. and Hu, J.: *VLSI physical design: from graph partitioning to timing closure*, Vol.312, Springer (2011).

[5] Hashimoto, A. and Stevens, J.: Wire routing by optimizing channel assignment within large apertures, *Proc. 8th Design Automation Workshop*, pp.155–169 (1971).

[6] Deutsch, D.N.: A dogleg channel router, *Proc. 13th Design Automation Conference*, pp.425–433 (1976).

[7] Yoshimura, T. and Kuh, E.S.: Efficient algorithms for channel routing, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Vol.CAD-1, No.1, pp.25–35 (1982).

[8] Ho, T.-T., Iyengar, S. and Zheng, S.-Q.: A general greedy channel routing algorithm, *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, Vol.10, No.2, pp.204–211 (1991).

[9] Takahashi, A. and Murata, H.: Three-Layer L-shaped Channel Routing Algorithm, *IPSJ Journal*, Vol.40, No.4, pp.1618–1625 (1999). (in Japanese).

[10] Takashima, Y., Takahashi, A. and Kajitani, Y.: Assignment of Intervals to Parallel Tracks with Minimum Total Cross-Talk, *IEICE Trans. Fundamentals*, Vol.E81-A, No.9, pp.1909–1915 (1998).

[11] Takahashi, A. and Kajitani, Y.: A Switch-Box Router 'BOX-PEELER' and Its Tractable Problems, *The Trans. IEICE*, Vol.E72, No.12, pp.1367–1373 (1989).

[12] Takashima, Y., Takahashi, A. and Kajitani, Y.: Routability of FPGAs with Extremal Switch-Block Structures, *IEICE Trans. Fundamentals*, Vol.E81-A, No.5, pp.850–856 (1998).

[13] Kajitani, Y.: Order of Channels for Safe Routing and Optimal Compaction of Routing Area, *IEEE Trans. Computer-Aided Design*, Vol.CAD-2, No.4, pp.293–300 (1983).

[14] Dai, W.M., Asano, T. and Kuh, E.S.: Routing Region Definition and Ordering Scheme for Building-Block Layout, *IEEE Trans. Computer-Aided Design*, Vol.CAD-4, No.3, pp.189–197 (1985).

[15] Wong, D.F. and Liu, C.L.: A New Algorithm for Floorplan Designs, *Proc. 23rd ACM/IEEE Design Automation Conference*, pp.101–107 (1986).

[16] Lee, C.Y.: An algorithm for path connection and its application, *IRE Trans. Electronic Computer*, Vol.EC-10, No.3, pp.346–365 (1961).

[17] Hart, P.E., Nilsson, N.J. and Raphael, B.: A Formal Basis for the Heuristic Determination of Minimal Cost Paths, *IEEE Trans. Systems Science and Cybernetics*, Vol.4, No.2, pp.100–107 (1968).

[18] Dees, W. and Karger, P.: Automated Rip-Up and Reroute Techniques, *Proc. 19th Design Automation Conference*, pp.433–439 (1982).

[19] Kohira, Y. and Takahashi, A.: CAFE router: A Fast Connectivity Aware Multiple Nets Routing Algorithm for Routing Grid with Obstacles, *IEICE Trans. Fundamentals*, Vol.E93-A, No.12, pp.2380–2388 (2010).

[20] Sato, S., Akagi, K. and Takahashi, A.: A Fast Length Matching Routing Pattern Generation Method for Set-Pair Routing Problem Using Selective Pin-Pair Connections, *IEICE Trans. Fundamentals*, Vol.E103-A, No.9, pp.1037–1044 (2020).

[21] Posser, G., Young, E.F., Held, S., Li, Y.-L. and Pan, D.Z.: Challenges and approaches in VLSI routing, *Proc. 2022 International Symposium on Physical Design*, pp.185–192 (2022).

Kazuya Taniguchi received his B.E. degree in information and communications engineering from Tokyo Institute of Technology, Tokyo, Japan, in 2022. He is currently a master course student of department of information and communications engineering in Tokyo Institute of Technology. His research interests include in VLSI layout design and combinational algorithms.

Satoshi Tayu received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1992, 1994, and 1997, respectively. From 1997 to 2003, he was a research associate in the School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan. He is currently an assistant professor with Department of Information and Communications Engineering, School of Engineering, Tokyo Institute of Technology. His research interests are in parallel computation. He is a member of IEICE and IPSJ.

Atsushi Takahashi received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from Tokyo Institute of Technology, Tokyo, Japan, in 1989, 1991, and 1996, respectively. He had been with the Tokyo Institute of Technology as a Research Associate from 1991 to 1997, and as an Associate Professor from 1997 to 2009 and from 2012 to 2015, and as a Professor from 2015. He had been with the Osaka University as an Associate Professor from 2009 to 2012. He is currently a Professor with Department of Information and Communications Engineering, School of Engineering, Tokyo Institute of Technology. His research interests include in VLSI layout design and combinational algorithms. He is a fellow of IEICE, a senior member of IEEE and IPSJ, and a member of ACM.

Mathieu Molongo received his B.E. degree in information and communications engineering from Telecon Paris Tech, Paris, France, in 2005. He is currently chief engineer at Jedat Inc. where he develops routing tools for LSI layout design.

Makoto Minami received his B.E., and M.E. degrees in precision engineering from the University of Tokyo, Tokyo, Japan, in 1995, and 1997, respectively. He is currently chief engineer at Jedat Inc. where he develops routing tools for LSI layout design.

Katsuya Nishioka received his B.S. degree in computer science and information engineering from Ibaraki University, Mito, Japan,in 1992. He is currently a Senior Manager, R&D, Jedat Inc., Japan. From 2010 to 2017 he was with Renesas Electronics Corp. as an engineer in EDA & Design Methodology Division.

(Recommended by Associate Editor: *Shiho Hagiwara*)