IEICE TRANS. FUNDAMENTALS, VOL. E81-A, NO. 5 MAY 1998

PAPER Special Section on Discrete Mathematics and Its Applications

# Routability of FPGAs with Extremal Switch-Block Structures

Yasuhiro TAKASHIMA†, *Nonmember*, Atsushi TAKAHASHI†, *and* Yoji KAJITANI†, *Members*

**SUMMARY** The switch-block architecture of FPGAs is discussed to see a good balance between programmable-switch resources and routability. For the purpose, FPGAs are assumed to have certain extremal structures, whose switch-blocks consist of *parallel* or *complete* switch-sets where a switch-set is a set of switches between two sides of the switch-block. A polynomial time detailed-routing algorithm for a given global-routing is presented if the switch-block consists of two or less parallel switch-sets or three that form a cycle. For other FPGAs, the corresponding decision problem is proved to be $\mathcal{NP}$-complete. A best compromise between switch resources and routability is offered.

***key words:*** *FPGA, switch-block, routability, detailed-routing*

## 1. Introduction

FPGAs are attracting popularity recent years as addressed from various aspects ([1] and elsewhere). With limited resources for routing, researches of architectures are to make them easier (1) to decide the routability of a given global-routing and/or (2) to design a routable global-routing.

There have been various considerations about the architecture of FPGAs [2], [3], [5]–[8]. They include some proposals of architectures and architecture-dependent strategies for global- and detailed-routing. Their contributions are demonstrated by experiments, or probabilistic considerations. Their considerations are within certain trade-offs between routability and reduction of routing resources. Thus, it is concluded that they are simply looking for a plausible compromise between "extremes." Therefore, studying FPGAs with some extremal architectures is essential for a profound understanding of roles of routing resources: programmable switches and wire segments. Such results are found in [3], [6], and [7]. In [3], the connection-block structure which has the minimum switches to meet any connection requirement is suggested. In [6] and [7], the routability in relation to the number of switches is discussed when the switch-blocks are of a certain structure.

This paper is a systematic study on the architecture of regular FPGAs whose switch-blocks are specifically extremal. All possible FPGAs of such extremal structures are listed and classified according to the structures. For each, the computational complexity of detailed routing for a given global routing is answered and a best compromise is concluded which will be a good principle in practical architecture design of FPGAs.

The rest of the paper is organized as follows. After preliminaries (Sect. 2), our problems are defined and main results are described in Sect. 3. In Sect. 4, a class of FPGAs whose routability problems are $\mathcal{NP}$-complete is provided. The proof of a lemma is left to Appendix. In Sect. 5, a polynomial time detailed-routing algorithm for the other classes of FPGAs is presented. Section 6 is the conclusion.

## 2. Preliminaries

Our FPGA model is a simplified one as shown in Fig. 1.

The structure of switch-blocks is assumed as follows (Fig. 2): The switch-block has the same number $w$ of terminals on four sides: the left, top, right, and bottom sides which are denoted by $L$, $T$, $R$, and $B$, respectively. The terminals on each side are labeled as $1, 2, \ldots, w$ from the top (for $L$ and $R$) and from the left (for $T$ and $B$). A *programmable switch* (simply a *switch* hereinafter) in a switch-block is located between a terminal on one side and that on another side. It connects the end terminals if activated. The set of terminals on

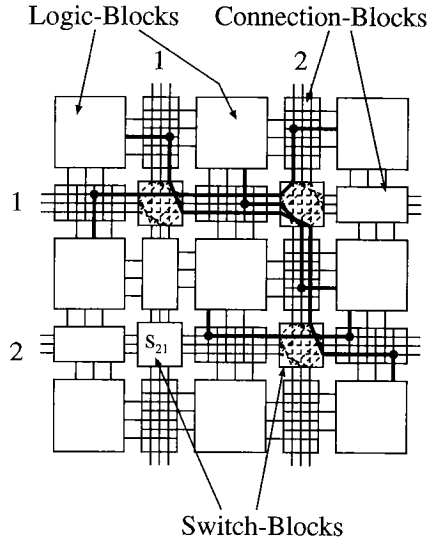



**Fig. 1** The FPGA model and a detailed routing.

Manuscript received September 16, 1997.

†The authors are with the Department of Electrical and Electronic Engineering, Tokyo Institute of Technology, Tokyo, 152–8552 Japan.

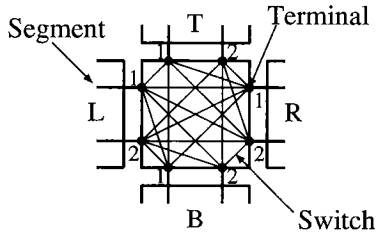



**Fig. 2** An extremally structured switch-block composed of either parallel or complete switch-sets. $s(T,B)$, $s(T,L)$, $s(B.R)$: parallel switch-sets. $S(T,R)$, $s(L,R)$, $s(L,B)$: complete switch-sets.

side $X$ ($= L, T, R$ or $B$) is denoted by $t_X$. The set of switches between $t_X$ and $t_Y$ is denoted by $s(X,Y)$ and called a *switch-set.*

A switch-set $s(X,Y)$ is called *complete* if every pair of a terminal on side $X$ and that on $Y$ has a switch, that is, $w^2$ switches are implemented. While a switch-set is called *parallel* if it comprises $w$ switches connecting the same labeled terminals (See Fig. 2). They are called the *extremal switch-sets* by the following sense: For the complete switch-set, any addition of a switch is not necessary for connection; For parallel switch-set, any remove of a switch fails in connecting $w$ terminals on the sides concerned.

As the alternative of the parallel switch-set, it is possible to consider the switch-set which is obtained by removing switches or/and permuting the terminals of the parallel switch-set. However, this paper neglects such switch-sets since they would make the discussion too complicated because of irregularity to give a simple guide-line for practical design.

There are six switch-sets in a switch-block. A switch-block is called an *extremally structured switch-block* if each switch-set is either complete or parallel. The variety of extremally structured switch-blocks is $2^6 = 64$. An FPGA whose switch-blocks are all extremally structured is called the *extremally structured FPGA.*

The FPGA considered in this paper is square consisting of $n \times n$ switch-blocks. Moreover each switch-block in the FPGA is square with $w \times w$ terminals and extremally structured. If all the switch-blocks in the FPGA are identical, the FPGA is said to be *regular.* Unless otherwise stated, we focus on the regular FPGAs.

The number of switches in a extremally structured switch-block is $n_s = \alpha w + (6-\alpha)w^2 = 6w^2 - \alpha(w^2 - w)$ where $\alpha$ is the number of parallel switch-sets of six switch-sets. Note that $n_s$ is considered to represent the amount of routing resources. It is expected that there is a general trade-off between $\alpha$ and routability such that the larger $\alpha$ is, the tighter the routability is.

The purpose of this paper is to get a definite idea of such trade-offs in FPGAs of all 64 possible extremal structures. A rough conclusion is that the minimum switch-block which keeps the routability problem in $\mathcal{P}$ is the switch-block in which $\alpha = 3$ such that three parallel switch-sets form a cycle. This is the architecture of the FPGA which the authors are recommending.

## 3. Problem Definitions and Theorems

The structure of *connection-block* is assumed as follows: Each connection-block has $w$ wire segments which connect the same labeled terminals on confronting sides of adjacent switch-blocks. Each wire segment in a connection-block can be connected to any I/O terminals on the incident logic-block. (See Fig. 3.)

The global-routing is the set of requests to the detailed-routing. For convenience to represent a global-routing, we define a connection-switch graph.

**Definition 1: Connection-Switch Graph (CS-graph)**
The *CS-graph* of an FPGA is an undirected graph which consists of the vertices corresponding the the connection-blocks and the edges correspond to the switch-sets.

By the model of the FPGA, the CS-graph is the union of $n^2$ complete graphs. Each complete graph corresponds to a switch-block and has four vertices which correspond to the incident connection-blocks. (See Fig. 4.)

A global routing of a signal is a network for which only terminals of logic-blocks, connection blocks, and switch-blocks are assigned to pass but no particular wire segments of the connection blocks nor switches of the switch-blocks are specified. Therefore, it is defined on the CS-graph.

**Definition 2: Global-Routing**
A global-routing of each signal is the set of connected subgraphs of the CS-graph. Each subgraph is called a *request-graph* and its edges are called *request-edges.*

Accordingly, the detailed-routing for each signal selects a switch from the switch-set corresponding to each request-edge. A network consisting of selected switches and wires incident to the switches is called a *switch-net* for the request-graph if the network is connected in the connection-block corresponding to each vertex of the request-graph.

**Definition 3: Detailed-Routing**
A detailed-routing is a set of switch-nets for given global-routing such that no two distinct switch-nets contain a common wire.

A global-routing is said to be *feasible* if there exists a corresponding detailed-routing or *infeasible* otherwise. A global-routing consisting two request-graphs is shown in Fig. 5. This global-routing is feasible since there is a detailed-routing as shown in Fig. 6.

We are concerned with the following problem.

**Definition 4: Routability Problem**

**Instance:** A global-routing.

**Question:** Is the global-routing feasible?

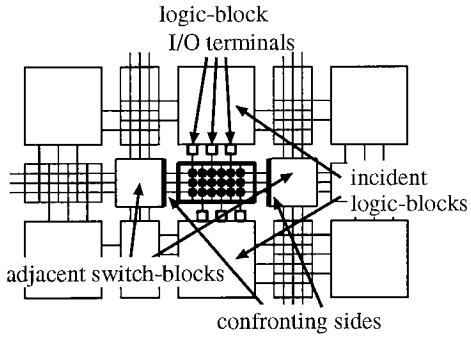



**Fig. 3** Adjacent switch-blocks and incident logic-blocks of a connection-block.

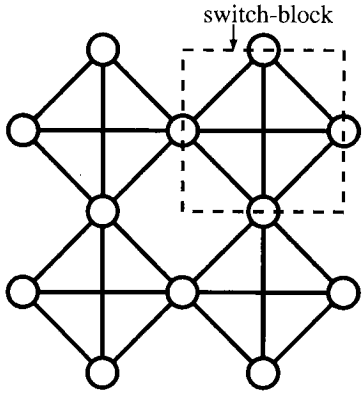



**Fig. 4** The CS-graph of FPGA in Fig. 1.

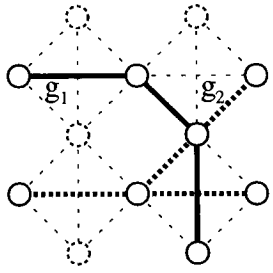



**Fig. 5** Request-graphs $g_1$ and $g_2$.

Since a switch-set is either complete or parallel, an FPGA is characterized only in terms of the parallel switch-sets in a switch-block. Therefore, we classify them into two classes by parallel switch-sets architecture.

**Definition 5: Classes "Disc" and "Conn" of FPGAs** An extremally structured regular FPGA belongs to class **Disc** if

1. the number of parallel switch-sets in a switch-block is at most 2 ($\alpha \leqq 2$), or
2. $\alpha = 3$ and parallel switch-sets form a cycle in a switch-block.

Class **Conn** is the set of other FPGAs, i.e. those whose switch-blocks satisfy

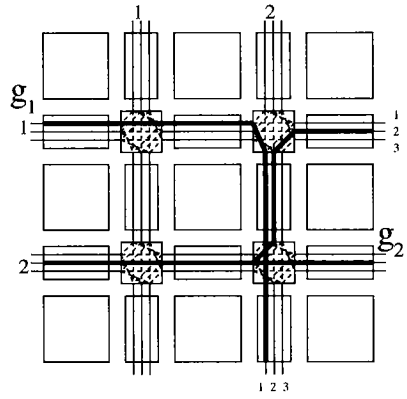



**Fig. 6** Detailed-routings of $g_1$ and $g_2$.

1. $\alpha \geqq 4$, or
2. $\alpha = 3$ and parallel switch-sets do not form a cycle in a switch-block.

An edge in the CS-graph corresponding to a parallel or complete switch-set is called a *paraS-edge* or *compS-edge*, respectively. The paraS-edge subgraph of a CS-graph is the graph obtained from the CS-graph by deleting all compS-edges.

**Lemma 1:**

1. The paraS-edge subgraph of an FPGA of **Disc** is disconnected.
2. The paraS-edge subgraph of an FPGA of **Conn** is connected.

**Proof:**

1. Let the FPGA consist of $p \times q$ switch-blocks. Assume that $T$ is a maximal tree of the paraS-edge subgraph. Since it contains at most two paraS-edges in the subgraph corresponding to a switch-block, the number of edges of $T$ is at most $2pq$. While the number of vertices of the paraS-edge subgraph is $2pq + p + q$. Therefore, $T$ cannot be a spanning tree.
2. Each four vertices corresponding to a switch-block are connected in paraS-edge subgraph. Then it is trivial that the paraS-edge subgraph itself is connected.

Their names **Conn** and **Disc** come from this property.

Main results are described in the following two theorems.

**Theorem 1:** For any FPGAs of **Conn**, the routability problem is $\mathcal{NP}$-complete.

**Theorem 2:** For any FPGAs of **Disc**, the routability problem is $\mathcal{P}$.

Proofs for Theorem 1 and Theorem 2 are given in Sects. 4 and 5, respectively.

## 4. FPGAs of Conn

First, consider an FPGA with $\alpha = 3$ such that parallel switch-sets contain no cycle. There are four possible distinct FPGAs taking symmetricity in consideration.

**Type 1:** $s(T, L)$, $s(T, R)$, and $s(B, L)$ are parallel.

**Type 2:** $s(T, B)$, $s(T, L)$, and $s(T, R)$ are parallel.

**Type 3:** $s(T, B)$, $s(T, L)$, and $s(B, R)$ are parallel.

**Type 4:** $s(T, B)$, $s(L, R)$, and $s(T, L)$ are parallel.

In Fig. 7, the paraS-edge subgraphs of Type 1 through Type 4 are shown. Note that a common feature of these graphs is that each is lead to a grid graph as shown Fig. 8 by shrinking edges enclosed by dotted cycles in Fig. 7.

Consider a special request such that each request-graph is a path consisting only of paraS-edges on these grid graphs. Since each signal must use the same labeled terminals and be disjoint from others, the routability problem for such an input is the problem: Determine if it is possible to assign those paths using $w$ labels such that any two paths that share a vertex have distinct labels. This is the problem: Grid-paths $w$-colorability, which will be formally defined and proved to be $\mathcal{NP}$-complete in Appendix.

Second, consider an FPGA with $\alpha = 4$. It is easy to see that each paraS-edge subgraph contains a subgraph which is either one of paraS-edge subgraphs of Type 1 through Type 4. Therefore, if we consider the request whose request-graphs are defined only on these edges, the problem contains the cases of $\alpha = 3$, thus $\mathcal{NP}$-complete. Similarly, since the cases of $\alpha = 5$ or 6 contain those of $\alpha = 4$, the problems of those cases are $\mathcal{NP}$-complete as well.

Thus we conclude that the Routability Problem for FPGAs of **Conn** is $\mathcal{NP}$-complete, and hence Theorem 1 has been proved.

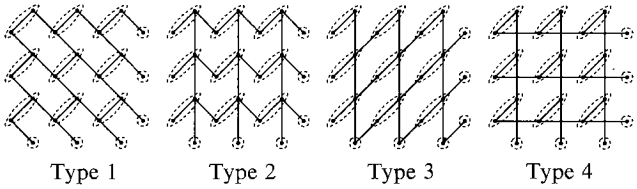



**Fig. 7** ParaS-edge subgraphs.

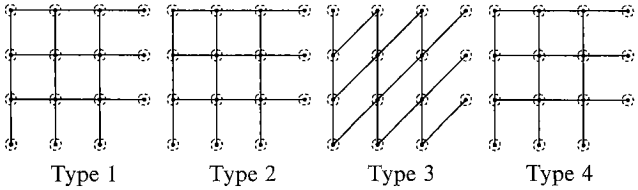



**Fig. 8** Shrunk paraS-edge subgraphs to grids.

## 5. FPGAs of Disc

We provide a linear time detailed-routing algorithm for the FPGAs of **Disc**.

Assume that a switch is selected from the switch-set corresponding to an a request-edge on the way of detailed-routing. Then, in the connection-block corresponding to its end vertex, a wire segment is determined used for the switch-net. Such a request-edge and its end vertices are said to be *fixed*. An unfixed edge incident to a fixed vertex is said to be *forced* by the vertex since the choice of a switch of the switch-block in order to form a switch-net is restricted.

Assume that there exists a forced paraS-edge in a request graph. If only one end vertex of the paraS-edge is fixed, the unique switch in the corresponding switch-set to be selected is specified in order to construct a switch-net for the request graph. While, if both end vertices are fixed to different labeled wire segments, it is impossible to select a switch to construct a switch-net for the request graph.

On the other hand, assume that there exists a forced compS-edge in a request graph. A switch can be selected even if two end vertices are fixed to different labeled wires.

**Lemma 2:** For an FPGA of **Disc**, a global-routing is feasible if and only if the following two conditions are satisfied:

**Vertex-cap:** For each vertex in CS-graph, there exist at most $w$ request graphs that contain the vertex.

**Cycle-cap:** For each paraS-edge cycle in CS-graph, there exist at most $w$ request graphs each of which contains at least one edge in the cycle.

**Proof:** The necessity of condition **vertex-cap** is trivial because a connection-block can convey at most $w$ distinct signals.

The necessity of condition **cycle-cap** is shown as follows. The length of a paraS-edge cycle is three. Therefore, if a request graph contains an edge of a cycle $c$, it means that a switch is used by the corresponding signal. The switch is connected to other two switches such that they together correspond to $c$. Therefore, three switches are occupied by one signal. While, the number of switches corresponding to $c$ is $3w$. Therefore, at most $w$ request-graphs exist.

The sufficiency will be proved by the algorithm in Fig. 9. Step 1 checks if these two conditions are satisfied. In Steps 3 and 4, detailed-routing is constructed.

It is possible to select the switch for each fixed paraS-edge, since only one end vertex is fixed. It is possible to select the switch for the other paraS-edge, since given global-routing satisfies **cycle-cap**.

On the other hand, it is possible to select the switch for each forced compS-edge, since there must exist the switch connecting to the fixed end vertex. It is possible

**Algorithm: detailed-route**
Input: A set of request-graphs

1. Return "the global-routing is infeasible" if either of conditions **vertex-cap** or **cycle-cap** is violated.
2. Give the priority to the switch-blocks from the top and then from the left (row-by-row).
3. For each switch-block according to the priority, select a switch from the parallel switch-sets in the switch-block as follows.
   - For each forced paraS-edge of request graphs contained in the switch-block, select the switch from the corresponding switch-set such that the switch is connecting to the wire where the end vertex are fixed.
   - For other paraS-edge of request graphs contained in the switch-block, select a switch from the corresponding switch-set such that the switch connects the wires where a vertex of the other request-graph is not fixed.
4. For each compS-edge of request graphs, select a switch from the corresponding switch-set such that the switch connects the wires where the end vertex is fixed if the end vertex is fixed, or where a vertex of the other request-graph is not fixed otherwise.

**Fig. 9** Detailed-routing algorithm.

to select the switch for other compS-edges, since given global-routing satisfies **vertex-cap**.

It is evident that the time complexity to test the two conditions is linear to the number of request edges. Hence, the algorithm works in linear time of the number of request-edges. Thus we conclude that the Routability Problem for the class **Disc** is $\mathcal{P}$, and Theorem 2 follows.

Note that the proof of the sufficiency uses only the fact that the length of the cycle consisting of paraS-edge is three and priority ordering in Step 2. Therefore Lemma 1 is extended to the theorem on general extremally structured FPGAs.

**Theorem 3:** For an extremally structured FPGA which is not necessarily regular, its routability problem is $\mathcal{P}$ if the length of any paraS-edge cycle of the CS-graph is at most 3.

**Proof:** We modify the priority ordering (Step 2). For the purpose, we first define graph $H$ as follows: A vertex corresponds to the switch-block which contains at least one paraS-edge. There exists an edge between two vertices if the corresponding switch-blocks are adjacent and the terminals of parallel switch-sets of these switch-blocks are connected by wire segments. Second, order the vertices of $H$ by Depth First Search starting with any vertex. Finally, give the priority ordering arbitrary to all the switch-blocks keeping the order given to the switch-blocks that contain parallel switch-sets.

$H$ does not contain cycles because $H$ is bipartite. So, there is no paraS-edge whose end vertices are fixed to different labeled terminals. Therefore, the detailed-routing algorithm with this modified priority ordering works.

## 6. Concluding Remarks

The architecture of practical FPGAs are very much sophisticated adopting the knowledges of expert's experiences and careful experiments. Still, theoretical aspects are important for further improvements to see reasons why such architectures have come out. Such a study often sets environmental restrictions to extremes. Our study is in this direction and the "rule of thumb" we obtained is useful: Three parallel switch-sets forming a cycle in each switch-block of regular FPGAs (*triangle FPGA*) is a good compromise between the routability and the switch resources.

In [7], Greedy Routing Architecture, called *GRA*, is provided. For FPGAs which consists of GRA switch-blocks, a global-routing is feasible if and only if only **vertex-cap** is satisfied. And there are $3.5w^2 + 2w$ switches in a switch-block. On the other hand, for our *triangle FPGA*, a global-routing is feasible if and only if **vertex-cap** and another condition, **cycle-cap** are satisfied. However, there are $3w^2+3w$ switches in a switch-block. So we can decrease the number of switches in a switch-block, adding simple condition, **cycle-cap**.

The future works are the study of trade-offs found in non-extremally structured FPGAs.

## Acknowledgment

Authors are grateful to Mr.Tomonori Izumi for his invaluable suggestion. This work has been supported in part by the Research Body of CAD21.

### References

[1] S.D. Brown, R.J. Francis, J. Rose, and Z.G. Vranesic, "Field-programmable gate arrays," Kluwer Academic Publishers, 1992.
[2] S. Brown, J. Rose, and Z.G. Vranesic, "An atochastic model to predict the routability of field-programmable gate arrays," IEEE Trans. Computer Aided Design, vol.11, pp.620–627, 1992.
[3] K. Fujiyoshi, Y. Kajitani, and H. Niitsu, "Design of optimum totally perfect connection-blocks of FPGA," ISCAS, 1994.
[4] M.R. Garey, D.S. Johnson, G.L. Miller, and C.H. Papadimitriou, "The complexity of coloring circular arcs and chords," SIAM J. Algebraic Discrete Methods, vol.1, pp.216–227, 1980.
[5] Y. Sun, T.C. Wang, C.K. Wong, and C.L. Liu, "Routing for symmetric FPGAs and FPICs," Proc. ACM/IEEE Design Automation Conf., pp.286–490, 1994.
[6] Y.L. Wu, S. Tsukiyama, and M. Marek-Sadowska, "Graph based analysis of FPGA routing," Proc. EURO-DAC with EURO-VHDL, pp.104–109, 1993.
[7] Y.L. Wu, D. Chang, M. Marek-Sadowska, and S. Tsukiyama, "Not necessarily more switches more routability," Proc. ASP-DAC, pp.579–584, 1997.

[8] K. Zhu, D.F. Wong, and Y.W. Chang, "Switch module design with application to two-dimensional segmentation design," Proc. IEEE/ACM Intl. Conf. Computer-Aided Design, pp.481–486, 1993.

## Appendix: $\mathcal{NP}$-completeness of Grid-paths $k$-colorability

Draw a circle on a plane. A straight line inside the circle whose endpoints are on the circle is called a *chord.* Let $C$ be a set of $n$ chords such that the endpoints of two chords are distinct. An assignment of colors to chords is called a *coloring* if no two chords that intersect on the plane are assigned the same color. A coloring using $k$ colors is called a *$k$-coloring.*

**Definition 6: Circle-chords $k$-colorability**

**Instance:** Set $C$ of chords, positive integer $k$.

**Question:** Is $C$ *$k$-colorable*, i.e., does there exist a function $f{:}C \rightarrow \{1, 2, \ldots, \mathrm{k}\}$ such that $f(a) \neq f(b)$ $(a, b \in C)$ whenever two chords $a$ and $b$ intersect?

It is known that this problem is $\mathcal{NP}$-complete [4].

**Definition 7: Grid-paths $k$-colorability**

**Instance:** Set $P$ of paths on a grid, positive integer $k$.

**Question:** Is $P$ *$k$-colorable*, i.e., does there exist a function $g{:}P \rightarrow \{1, 2, \ldots, \mathrm{k}\}$ such that $g(p) \neq g(q)$ $(p, q \in P)$ whenever two paths $p$ and $q$ share a grid point?

The objective is to prove that Grid-paths $k$-colorability is $\mathcal{NP}$-complete. We provide a polynomial time reduction of Circle-chords $k$-colorability to the problem.

Let $C$ be a chord set. By assigning labels $1, 2, \ldots, 2n$ to endpoints of chords clockwise along the circle, a chord is denoted by an ordered pair of labels of its endpoints as $(i, j)$ where $i < j$. A chord $(i, j)$ in $C$ is said to be *minimal* in $C$ if there is no chord $(x, y) \in C$ such that $i < x < y < j$. Similarly, a chord $(i, j) \in C$ is *maximal* in $C$ if there is no chord $(x, y) \in T$ such that $x < i < j < y$. Classify all chords into five classes:

$$C_T = \{(i,j) | i, j \leqq n\};$$
$$C_- = \{(i,j) | j > n, i + j < 2n + 1\};$$
$$C_0 = \{(i,j) | i + j = 2n + 1\};$$
$$C_+ = \{(i,j) | i \leqq n, i + j > 2n + 1\};$$
$$C_B = \{(i,j) | i, j > n\}.$$

We are going to construct a path set $P$ on a grid $G$ of size $|C| \times |C|$ which will satisfy the following properties.

1. The grid points on the top row and bottom row are assigned consecutively with labels $1, 2, \ldots, n$ and $2n, 2n-1, \ldots, n+1$, both from the left, respectively.
2. A path corresponding to a chord $(i, j) \in C$ connects points labeled $i$ and $j$.
3. Two paths corresponding to chords $(i, j)$ and $(p, q)$ intersect if and only if $(i, j)$ and $(p, q)$ intersect.

A procedure to construct such $P$ is described in the following.

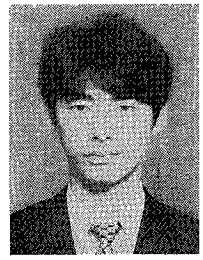
1. Find the chord set $C_1$ which consists of all minimal chords in $C_T$. For each chord $(i, j)$ in $C_1$, construct a path in the grid to connect the points labeled $i$ and $j$ in the top row. Then, delete $C_1$ from $C_T$. Find the chord set $C_2$ which consists of all minimal chords in the resultant $C_T$. For each chord in $C_2$ construct a path using two columns and the second row. Continue the process until $C_T$ is exhausted. Next take $C_-$ and continue the same process but using a row which is next to the preceding row, until to exhaust $C_-$.
2. Construct paths corresponding to $C_0$ by straight vertical lines.
3. Then take the $C_+$ and $C_B$. Continue the similar process but finding the chord set which consists of all maximal chords.

It is trivial that $P$ satisfies property 1 and 2. We show that $P$ satisfies the property 3. It is trivial that if two chords intersect, the corresponding paths share a point.

Next consider two chords $(i, j)$ and $(p, q)$ which do not intersect. Assume that both belong to $C_T$. If $j < p$, it is trivial that the paths do not intersect by construction. Otherwise, we assume without loss of generality that $i < p$ and $q < j$. The horizontal segment of the path $(p, q)$ is above that of $(i, j)$ by construction. Thus, these two paths do not share a grid point.

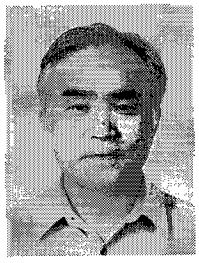
Other cases can be verified analogously.

Thus, a solution for the problem Grid-paths $k$-colorability is a solution of the problem Circle-chords $k$-colorability. This completes the proof of Grid-paths $k$-colorability being $\mathcal{NP}$-complete.



**Yasuhiro Takashima** received his B.E. and M.E. degrees in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1993 and 1995, respectively. He is currently a Ph.D. candidate of electronic engineering at the Tokyo Institute of Technology. His research interests are in combinatorial algorithms applied to VLSI layout design.

**Atsushi Takahashi** received his B.E., M.E., and D.E. degrees in electrical and electronic engineering from the 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 has been an associate professor since 1997 in the Department of Electrical and Electronic Engineering. His research interests are in VLSI layout design and combinatorial algorithms. He is a member of the Information Processing Society of Japan and IEEE.



**Yoji Kajitani** received his B.E., M.E. and D.E. degrees from the Tokyo Institute of Technology, Tokyo, Japan, all in electrical engineering, in 1964, 1966 and 1969, respectively. He has been a professor in the Department of Electrical and Electronic Engineering at the Tokyo Institute of Technology since 1985, and has been a professor at the Japan Advanced Institute of Science and Technology from 1992 to 1995. His current research interests are in combinatorial algorithms applied to VLSI layout design. He was awarded IEEE Fellowship in 1992.