Monotonic Parallel and Orthogonal Routing for Single-Layer Ball Grid Array Packages

Yoichi Tomioka Atsushi Takahashi

Department of Communications and Integrated Systems, Tokyo Institute of Technology
2–12–1–S3–58 Ookayama, Meguro-ku, Tokyo, 152–8550 Japan
Tel: +81-3-5734-2665
Fax: +81-3-5734-2902
e-mail: {yoichi,atsushi}@lab.ss.titech.ac.jp

Abstract— In this paper, we give the necessary and sufficient condition that all nets can be connected by monotonic routes when a net consists of a finger and a ball and fingers are on the two parallel boundaries of the Ball Grid Array package, and propose a monotonic routing method based on this condition. Moreover, we give a necessary condition and a sufficient condition when fingers are on the two orthogonal boundaries, and propose a monotonic routing method based on the necessary condition.

I. Introduction

Ball Grid Array (BGA) packages as shown in Fig. 1, in which I/O pins are placed in a grid array pattern, realize a number of connections between chips and the printed circuit board (PCB). Bonding fingers are connected to chips, and solder balls are I/O pins of the package in a grid array pattern. Since the structure of BGA packages is simple, many routes can be realized in few layers in the packages if connection requirements and routing patterns are suitable for the structure. In current package routing design, the designer generates satisfactory routing patterns by using the properties of connection requirements effectively. But it takes much time for large packages since the huge number of routes needs to be realized. So, the demand for automation of package routing is increasing. In this paper, we consider routing for a single-layer BGA package as the first step for BGA packages routing.

In the literature on for planar routing, there are a lot of problem formulations and approaches. For example, problem formulations for single-row and double-row routing, where terminals are placed on single-row and double-row, are proposed in [1] and [2], respectively. Though these problem formulations are similar to problems for single-layer BGA packages, approaches for them are not enough to obtain satisfactory routes for BGA packages. Actually, many parts of the routing process for BGA packages are realized manually with support tools.

In order to obtain a satisfactory routing pattern, the analysis of manual routing patterns is necessary. In the routing pattern by manual, though routes may snake, most of them do not go back. The route which do not go back are said to be monotonic. In monotonic routing patterns, it is expected that the total wire length tends to be small, and it is easy to decide the route of each net. But there exists a netlist that cannot be realized by monotonic routes in one layer with any design rule. There also exists a netlist in which a design rule may be satisfied if non-monotonic routes are allowed. In these cases, non-monotonic routes are needed. Though we aim to realize nets in one layer under a certain design rule, in this paper we propose an approach in which all nets are realized by monotonic routes. The obtained monotonic routes will be an initial solution in iterative improvement to satisfy the design rule.

In literatures for BGA package, several approaches focusing on monotonic routes were proposed. The first approach for single-layer BGA packages was proposed in [3] and it was improved in [4]. Their approach generates optimal uniform distribution of wire by generating connection requirements. An approach for 2-layer BGA packages was proposed in [5]. It is given connection requirements, and optimizes the total wire length and the wire congestion by improving via assignment.

Also, several approaches considering non-monotonic routes were proposed. The approach for multilayer Pin Grid Array (PGA) and Ball Grid Array packages were proposed in [6] and [7], respectively. They assign each net to a layer, and realize nets in respective layer.

All of them divide the package into several sectors, and nets are realized within each sector. Basically, each sector consists of bonding fingers on the same boundary of the package and solder balls, and a net in each sector consists of a bonding finger and a solder ball. Namely, their approach cannot be applied if it is impossible to divide the package into such sectors. So, we propose an approach for the region consisting of solder balls and bonding fingers on two boundaries of the package as shown in Fig. 2.

Section II introduces routing model, and gives some definitions for analysis. Connection requirements are the set of nets and are called a netlist. A parallel netlist, in which bonding fingers are on two parallel boundaries of the package, is shown in Fig. 2(a). In section III, we give

0-7803-9451-8/06/$20.00

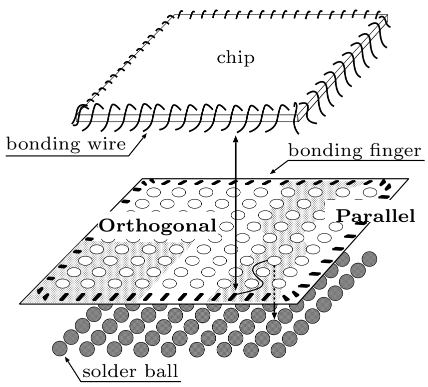
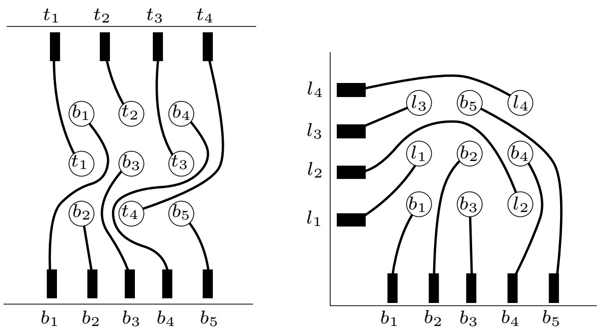


Fig. 1. A Ball Grid Array package



(a) Parallel Netlist **(b) Orthogonal Netlist**

Fig. 2. Monotonic Netlists Decision Problem

the necessary and sufficient condition that all nets in the parallel netlist can be realized by monotonic routes, and propose a monotonic routing method based on this condition. An orthogonal netlist, in which bonding fingers are on two orthogonal boundaries of the package, is shown in Fig. 2(b). In section IV, we give a necessary condition and a sufficient condition that all nets in the orthogonal netlist can be realized by monotonic routes, and propose a monotonic routing method based on the proposed necessary condition. There may exist more than one monotonic routing pattern that corresponds to a parallel netlist and an orthogonal netlist, respectively. How to select one among them that meets the design rule, if it exists, is in our future works. Since it is not guaranteed that our routing method for orthogonal netlists completes routing, we implement our method for orthogonal netlists with C++ language, and applied it to orthogonal netlists in section V. Section VI concludes this paper.

II. Preliminary

A. Definitions

In this paper, we assume that the BGA package has connection requirements between bounding fingers placed on boundaries of the package and solder balls placed in a grid array pattern. A solder ball, which we will refer to as a ball, is an I/O pin of the package and is connected to the PCB. A bonding finger, which we will refer to as a finger, is connected to the chip by a bonding wire. In this paper, we assume that all nets are two-terminal nets connecting a finger to a ball. A netlist is the set of such nets and is represented by $\mathbf{N}$. We refer to a finger placed on a bottom boundary of the package as a bottom finger, and refer to a net which consists of a bottom finger and a ball as a bottom net. Similarly, a top net and a left net are defined. Bottom nets and top nets are labeled according to the order of fingers from the left to the right as $b_1, b_2, b_3, \ldots$ and $t_1, t_2, t_3, \ldots$, respectively. Left nets are labeled according to the order of fingers from the bottom to the top as $l_1, l_2, l_3, \ldots$. Let $\mathbf{B}, \mathbf{T}$ and $\mathbf{L}$ be the sets of bottom, top and left nets, respectively.

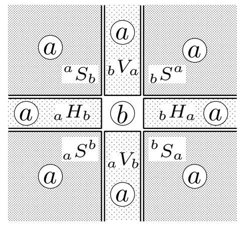


Fig. 3. Relationship between a and b

We define the relation between nets according to their ball positions as shown in Fig. 3. Let (x_a, y_a) and (x_b, y_b) be the coordinates of the balls of nets a and b, respectively. The relation between a and b is defined as follows.

- If $x_a < x_b, y_a = y_b$ then a is said to be to the left of b and the relation is represented by ${}_aH_b$.
- If $x_a = x_b, y_a < y_b$ then a is said to be below b and the relation is represented by ${}_aV_b$.
- If $x_a < x_b, y_a < y_b$ then a is said to be to the lower-left of b and the relation is represented by ${}_aS^b$.
- If $x_a < x_b, y_a > y_b$ then a is said to be to the upper-left of b and the relation is represented by ${}^aS_b$.
- ${}_bH_a$, ${}_bV_a$, ${}_bS^a$ and ${}^bS_a$ are defined symmetrically.

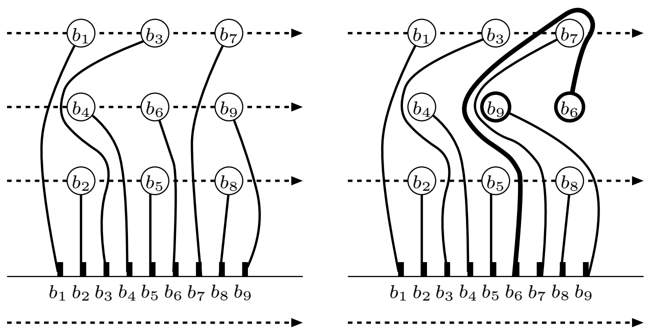
B. Order Graphs

We use some order graphs where a vertex v corresponds to a net $v \in \mathbf{N}$. The edge from a vertex u to a vertex v is represented by the ordered pair (u, v). In this paper, every order graph has edges corresponding to the order of fingers in each boundary. E_f^b, E_f^t and E_f^l are the sets of edges corresponding to the order in bottom, top and left boundaries, respectively. Formally, they are given as follows:

$$E_f^b = \{(b_i, b_j) \mid b_i, b_j \in \mathbf{B}, i < j\},$$
$$E_f^t = \{(t_i, t_j) \mid t_i, t_j \in \mathbf{T}, i < j\},$$
$$E_f^l = \{(l_i, l_j) \mid l_i, l_j \in \mathbf{L}, i < j\}.$$

C. Monotonic Routes

A boundary, where the finger of a net is placed, is called the finger boundary of the net. For example the finger



(a) All routes are monotonic (b) $R(b_6)$ is non-monotonic

Fig. 4. Monotonic and non-monotonic routes

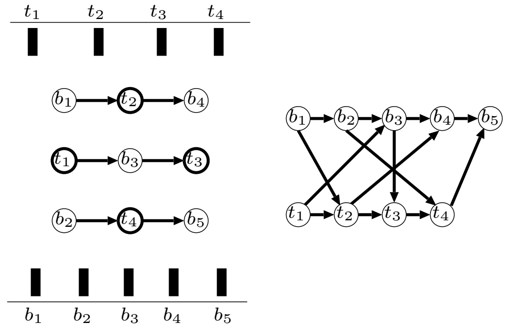


Fig. 5. MPN Decision Problem

boundary of b_1 in Fig. 2(a) is the bottom boundary. In this paper, a monotonic route and a non-monotonic route are defined as follows:

Definition 1 *If the route from a finger to a ball intersects any straight lines running parallel with the finger boundary at most once, then the route is said to be monotonic. Otherwise the route is said to be non-monotonic.*

Let $R(v)$ be the route of a net $v \in \mathbf{N}$. All routes are monotonic in Fig. 4(a), but $R(b_6)$ is non-monotonic in Fig. 4(b).

If all nets in a netlist can be realized by monotonic routes without intersecting each other, the netlist is said to be monotonic. A netlist is said to be single if the fingers of nets in the netlist are placed on the same boundary.

Consider that nets consist of bottom fingers and balls as shown in Fig. 4. A single netlist is monotonic if and only if nets on each row are in increasing order. Since a netlist in Fig. 4(a) satisfies this condition, it is monotonic. On the other hand, either $R(b_6)$ or $R(b_9)$ is non-monotonic in Fig. 4(b) since b_6 and b_9 are in decreasing order. A monotonic routing pattern for a monotonic single netlist, in which all routes are monotonic, is unique. Similar observations are found in [3, 4, 5].

Whether a single netlist is monotonic is decided by the order graph G_S. $E(G_S)$ consists of E_f^b and edges that correspond to the order of nets on each row. For example in Fig. 4(a), G_S has edges $(b_2, b_5), (b_2, b_8)$ and (b_5, b_8) corresponding to the bottom row. Similarly, G_S has edges for other rows. Clearly, G_S is cyclic if and only if there exist nets on a row which are in decreasing order, such as b_6 and b_9 in Fig. 4 (b).

III. Parallel Netlists

A parallel netlist is a netlist in which fingers are placed on the two parallel boundaries of the package. In this section, we analyze parallel netlists.

A. Monotonic Parallel Netlists

The Monotonic Parallel Netlist (MPN) Decision Problem is defined as follows:

Definition 2 *MPN Decision Problem*
Input:
A parallel netlist.
Question:
Is it possible to realize all connection requirements by monotonic routes?

An example of MPN Decision Problem is given in Fig. 2(a). In this case, $\mathbf{N} = \mathbf{B} \cup \mathbf{T}$. The necessary and sufficient condition for being monotonic is that nets on each row are in increasing order without distinguishing bottom and top nets. This condition is represented by the order graph G_P. $E(G_P)$ consists of E_f^b, E_f^t and edges corresponding to the order of nets on each row. In Fig. 5, G_P has edges $(b_2, t_4), (b_2, b_5)$ and (t_4, b_5) corresponding to the bottom row. Similarly, G_P has edges for other rows. In Fig. 5, the transitive edges like (b_2, b_5) are omitted. A parallel netlist is monotonic if and only if G_P is acyclic.

Theorem 1 *A parallel netlist is monotonic if and only if the order graph G_P is acyclic, where $E(G_P) = E_f^b \cup E_f^t \cup E_p$ and $E_p = \{(x, y) \mid x, y \in \mathbf{N}, {}_xH_y\}$.*

Proof. If the order graph G_P is acyclic, then an order can be obtained by G_P. A monotonic routing pattern can be realized according to the order as shown in section III.B. Conversely, consider that G_P has a cycle C. If C consists of only bottom nets, then non-monotonic routes are needed since it means that bottom nets are in decreasing order on a row. The same discussion is possible for top nets. So we assume that C consists of bottom nets and top nets. Without loss of generality, we assume that C has (b_j, t_p) and (t_q, b_i), where $b_i, b_j \in \mathbf{B}$ $(i < j)$ and $t_p, t_q \in \mathbf{T}$ $(p < q)$. Since ${}_{b_j}H_{t_p}$ and ${}_{t_q}H_{b_i}$, non-monotonic routes are needed by at least one of them as shown in Fig. 6. □

B. A Parallel Routing Method

A partial order is defined by G_P, and some orders are obtained by the partial order. This order corresponds to an order such that the sources in G_P are removed one by one. For example, the following order

$$t_1 \to b_1 \to b_2 \to t_2 \to b_3 \to b_4 \to t_3 \to t_4 \to b_5$$

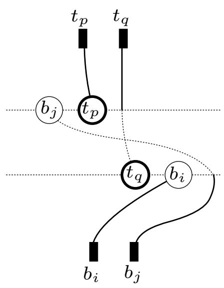


Fig. 6. ${}_{b_j}H_{t_p} \wedge {}_{t_q}H_{b_i}$

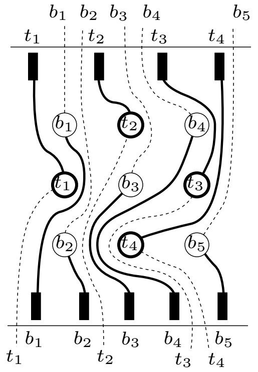


Fig. 7. An example of monotonic routing pattern for MPN

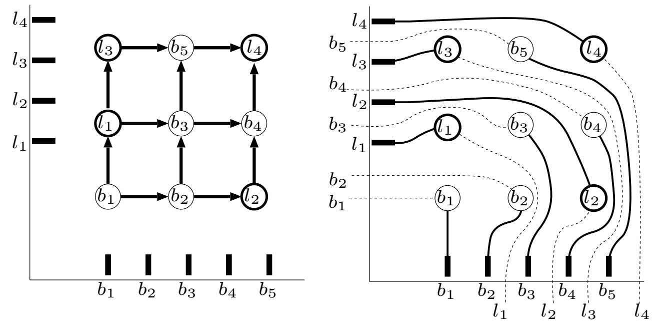


Fig. 8. Examples of MON Decision Problem and it's monotonic routing pattern

is obtained for Fig. 5. According to the obtained order, we put virtual fingers on bottom boundary of the package for top nets and put virtual fingers on top boundary for bottom nets. All nets can be realized because we can connect a finger to its virtual finger via its ball by monotonic route one by one from the left. An example of solution are given in Fig. 7.

For a monotonic parallel netlist, there are several monotonic routing patterns since the order obtained by G_P is not unique in general. The selection of the order that reduces the density is in our future work.

IV. Orthogonal Netlists

An orthogonal netlist is a netlist in which fingers are placed on the two orthogonal boundaries of the package. In this section, we analyze orthogonal netlists.

A. Monotonic Orthogonal Netlists

The Monotonic Orthogonal Netlist (MON) Decision Problem is defined as follows:

Definition 3 *MON Decision Problem*
Input:
An orthogonal netlist.
Question:
Is it possible to realize all connection requirements by monotonic routes?

An example of MON Decision Problem is given in Fig. 2(b). In this case, $\mathbf{N} = \mathbf{B} \cup \mathbf{L}$.

A.1 A Sufficient Condition

An orthogonal netlist is monotonic if the order graph G_s, which has edges corresponding to the order of nets on each row and column without distinguishing bottom and left nets, is acyclic. According to the order given by G_s, we put virtual fingers. Nets are realized by connecting each finger to its virtual finger via its ball from the lower left. Examples of MON Decision Problem and it's monotonic routing pattern are given in Fig. 8.

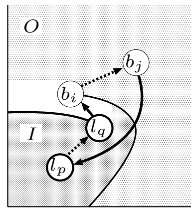


Fig. 9. A directed graph G_R

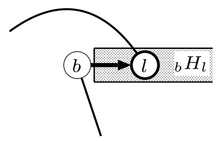


Fig. 10. A constraint between two nets

Theorem 2 *An orthogonal netlist is monotonic if the order graph G_s is acyclic, where $E(G_s) = E_f^b \cup E_f^l \cup E_s$ and $E_s = \{(x, y) \mid x, y \in \mathbf{N}, {}_xH_y \vee {}_xV_y\}$.*

If b_2 in the second column in Fig. 8 is swapped for b_3, G_s becomes cyclic. But, the netlist is monotonic since a monotonic routing pattern for the netlist is given in Fig. 2(b). So, this condition is not a necessary condition.

A.2 A Necessary Condition

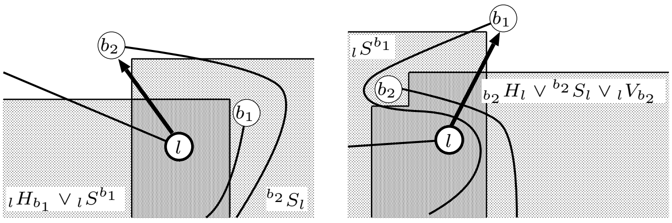
A route can be regarded as the set of points. Let b be a bottom net and v be a net. If there exists a point (x_b, y_b) on $R(b)$ and a point (x_v, y_v) on $R(v)$ such that $x_b < x_v$ and $y_b = y_v$, then $R(v)$ is said to be to the right of $R(b)$. In other words, $R(v)$ is said to be to the right of $R(b)$ if there exists a point on $R(v)$ which is to the right of $R(b)$. Similarly, $R(v)$ for net v is said to be above $R(l)$ for left nets if there exists a point on $R(v)$ which is above $R(l)$.

Theorem 3 *Let G_R be the directed graph constructed for a routing pattern, where the vertices correspond to the nets, and edge set is defined as follows:*

- An edge (b, v) $(b \in \mathbf{B}, v \in \mathbf{N})$ exists if and only if $R(v)$ is to the right of $R(b)$.
- An edge (l, v) $(l \in \mathbf{L}, v \in \mathbf{N})$ exists if and only if $R(v)$ is above $R(l)$.

If the routing pattern is monotonic, then G_R is acyclic.

Proof. Consider that G_R is cyclic. Let C be a cycle in G_R. The cycle cannot consist of only bottom nets or only left nets since there is no non-monotonic route. So, there is an edge from a bottom net to a left net and an



(a) $^{b_2}S_{b_1}$ (b) $_{b_2}S^{b_1}$

Fig. 11. Constraints between three nets

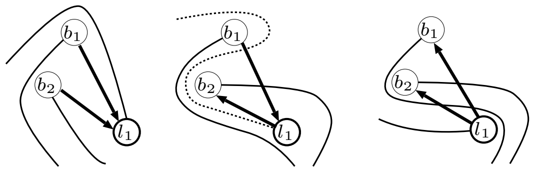


Fig. 12. An example of alternative constraints $(l_1, b_1) \bigoplus (b_2, l_1)$

edge from a left net to a bottom net. Without loss of generality, we assume that C includes (b_j, l_p) and (l_q, b_i), and that $R(b_i)$ is above $R(l_q)$, where $b_i, b_j \in \mathbf{B}$ $(i \leq j)$ and $l_p, l_q \in \mathbf{L}$ $(p \leq q)$. See Fig. 9. Since all routes are monotonic, the ball of b_j and $R(b_j)$ need to exist in region O shown in Fig. 9. Similarly, the ball of l_p and $R(l_p)$ need to exist in region I. Therefore, $R(l_p)$ is not to the right of $R(b_j)$, since $x_i < x_o$ if $y_i = y_o$ where (x_i, y_i) and (x_o, y_o) are points in region I and O, respectively. However, G_R has the edge (b_j, l_p). It contradicts definition of $E(G_R)$. So, G_R is acyclic if all routes are monotonic. □

G_R is not defined when a monotonic routing pattern is not given. However, depending on the relationship between bottom net b and left net l, there are cases such that it is decided that $R(l)$ is to the right of $R(b)$ or $R(b)$ is above $R(l)$ in any monotonic routing pattern. In such cases, (b, l) or (l, b) exist in G_R for any monotonic routing pattern. A graph where only such edges exist is the graph obtained from G_R by removing some of edges. Therefore, we consider such order graph G_n. Clearly, an orthogonal netlist is not monotonic if G_n is cyclic.

For example, if $R(b)$ and $R(l)$ are monotonic and $_bH_l$ as shown in Fig. 10, then an edge (b, l) is in G_n since $R(l)$ is always to the right of $R(b)$. Similarly, $R(b_2)$ is always above $R(l)$ if three balls of nets b_1, b_2 and l are placed as shown in Fig. 11(a). In Fig. 11(b), $R(b_1)$ and $R(b_2)$ are always above $R(l_1)$.

A necessary condition is given focusing on two or three nets. G_n can be constructed by necessary conditions for being monotonic.

Theorem 4 *An orthogonal netlist is not monotonic if the order graph G_n is cyclic, where $E(G_n) = E_f^b \cup E_f^l \cup E_h \cup E_v \cup E_1^b \cup E_1^l \cup E_2^b \cup E_2^l$,*

$E_h = \{(b, x) \mid {}_bH_x\}$,
$E_v = \{(l, x) \mid {}_lV_x\}$,
$E_1^b = \{(l, b_j) \mid {}^{b_j}S_{b_i} \wedge ({}_lH_{b_i} \vee {}_lS^{b_i}) \wedge {}^{b_j}S_l\}$,
$E_1^l = \{(b, l_j) \mid {}^{l_i}S_{l_j} \wedge ({}_bV_{l_i} \vee {}_bS^{l_i}) \wedge {}^bS_{l_j}\}$,
$E_2^b = \{(l, b_i) \mid {}_{b_j}S^{b_i} \wedge {}_lS^{b_i} \wedge ({}_{b_j}H_l \vee {}^{b_j}S_l \vee {}_lV_{b_j})\}$,
$E_2^l = \{(b, l_i) \mid {}_{l_j}S^{l_i} \wedge {}_bS^{l_i} \wedge ({}_bH_{l_j} \vee {}^bS_{l_j} \vee {}_{l_j}V_b)\}$,
where $x \in \mathbf{N}$, $b, b_i, b_j \in \mathbf{B}$, $l, l_i, l_j \in \mathbf{L}$ and $i < j$.

In addition, there are alternative constraints. When three balls of nets b_1, b_2 and l_1 are placed as shown in Fig. 12, $R(l_1)$ is non-monotonic if $R(l_1)$ is to the right of $R(b_1)$ and below $R(b_2)$. So $R(l_1)$ is either below or to the right of $R(b_1)$ and $R(b_2)$. Therefore, either (l_1, b_1) or (b_2, l_1) should exist in G_n. These constraint is represented by $(l_1, b_1) \bigoplus (b_2, l_1)$.

There exists an alternative constraint $(l, b_i) \bigoplus (b_j, l)$ if $({}_{b_j}V_{b_i} \vee {}_{b_j}S^{b_i}) \wedge {}^{b_i}S_l \wedge {}^{b_j}S_l$, where $b_i, b_j \in \mathbf{B}$ $(i < j)$ and $l \in \mathbf{L}$. Similarly, alternative constraints for two left nets and a bottom net are defined.

Assume that there is an alternative constraint $(l, b_i) \bigoplus (b_j, l)$. If G_n becomes cyclic when (l, b_i) is added in G_n, (b_j, l) should be selected. But if G_n does not become cyclic for either edge, then it is not easy to decide which is to be selected. When there are some alternative constraints, an orthogonal netlist is not monotonic if G_n are cyclic for all combinations of alternative constrains. This constraints should be analyzed thoroughly in our future work since the number of combinations is exponential for the number of alternative constraints.

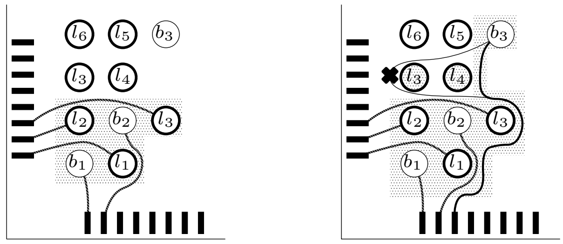
B. An Orthogonal Routing Method

An initial order graph is constructed by necessary conditions without alternative constraints. If decision is possible for an alternative constraint, the corresponding edge is added. The order of nets is determined by G_n by removing the source one by one. The order graph is updated if removal of the source forces an alternative constraint and the decision is possible for it. In this method, the updated order graph might become cyclic depending on the selection of source.

According to the obtained order, fingers are connected to balls by monotonic routes one by one from the lower left. Formally, routing of a bottom net is defined as follows: A ball is said to be connected if its route is completed. Otherwise, a ball is said to be unconnected. Let b be a bottom net. $R(b)$ passes as the left as possible on condition that $R(b)$ passes to the right of the unconnected left net balls in the lower-left region of b and connected balls. For example, consider $R(b_3)$ in Fig. 13. $R(l_3)$ and $R(l_4)$ become non-monotonic if $R(b_3)$ passes to the left of them. Therefore, $R(b_3)$ needs to avoid balls of nets l_3 and l_4 as shown in Fig. 13(b). Similarly, routes of left nets can be decided.

V. Experiments and Results

We implemented our method for orthogonal netlists with C++ language and applied it to monotonic orthogonal netlists since it is not guaranteed that our method completes routing. Monotonic orthogonal netlists are



(a) Before b_3 is realized. (b) After b_3 is realized.

Fig. 13. An example of making routes

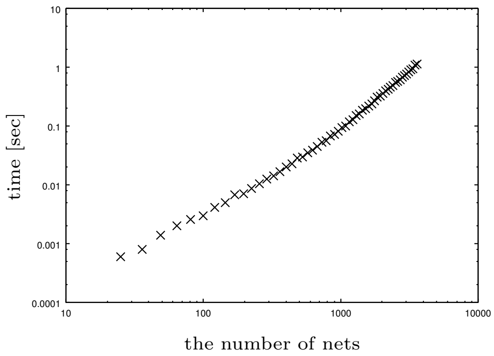


Fig. 14. The relationship between the number of nets and time for monotonic orthogonal netlist

generated by relaxing the sufficient condition in section IV.A.1.

In this experiment, we used the order graph G, where $E(G) = E_f^b \cup E_f^l \cup E_h \cup E_v \cup E_1^b \cup E_1^l$. We applied it to problems of 56 size from 5×5 to 60×60. 100 patterns were generated in each size. Two instances in 44×44, 45×45 could not be completed, since G became cyclic due to the alternative edges added in routing process. The others were completed, and monotonic routing patterns were generated. The graph between the number of nets and average execution time are shown in Fig. 14 , and an example of output is shown in Fig. 15. The algorithm generates a monotonic routing pattern within 1 second even for 3000 nets problem. The practical algorithm will be obtained if the density is taken into account in order selecting.

VI. Conclusion

We gave the necessary and sufficient condition for parallel netlist being monotonic, and proposed a routing method for monotonic parallel netlists based on this condition. Moreover we gave a necessary condition and a sufficient condition for orthogonal netlists being monotonic, and proposed a routing method for monotonic orthogonal netlists based on the necessary condition.

As our future work, we need to investigate alternative constraints and whether there are constraints between four or more nets. Routing methods that take routing density into consideration should be proposed. Moreover, the method taking non-monotonic routes into account should be proposed. In the method, the monotonic routes obtained by our proposed method is used as an initial solution and is improved iteratively to satisfy the design rule.

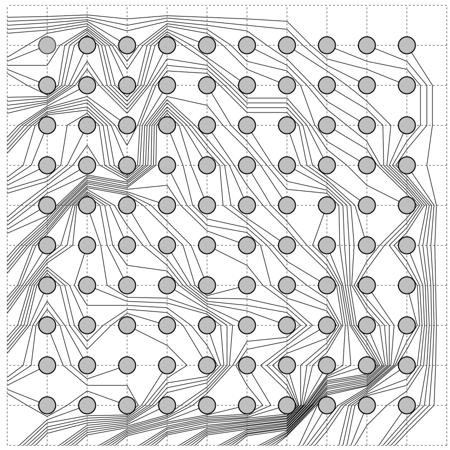


Fig. 15. An example of output (100 nets)

References

[1] E. S. Kuh, T. Kashiwabara, and T. Fujisawa, "On Optimum Single-Row Routing," *IEEE Transactions on Circuits and Systems*, vol. CAS-26, no. 6, pp. 361–368, 1979.

[2] S. Tsukiyama and E. S. Kuh, "Double-Row Planar Routing and Permutation Layout," *networks*, vol. 12, no. 3, pp. 287–316, 1982.

[3] M.-F. Yu and W. W.-M. Dai, "Single-Layer Fanout Routing and Routability Analysis for Ball Grid Arrays," in *Proceedings of International Conference Computer-Aided Design*, pp. 581–586, 1995.

[4] S. Shibata, K. Ukai, N. Togawa, M. Sato, and T. Ohtsuki, "A BGA Package Routing Algorithm on Sketch Layout System," *The journal of Japan Institute for Interconnecting and Packaging Electronic Circuits*, vol. 12, no. 4, pp. 241–246, 1997. (In Japanese).

[5] Y. Kubo and A. Takahashi, "A Global Routing Method for 2-Layer Ball Grid Array Packages," in *Proceedings of ACM International Symposium on Physical Design*, pp. 36–43, April 3-6 2005.

[6] C.-C. Tsai, C.-M. Wang, and S.-J. Chen, "NEWS: A Net-Even-Wiring System for the Routing on a Multilayer PGA Package," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 2, pp. 182–189, 1998.

[7] S.-S. Chen, J.-J. Chen, C.-C. Tsai, and S.-J. Chen, "An Even Wiring Approach to the Ball Grid Array Package Routing," in *Proceedings of International Conference on Computer Design*, pp. 303–306, 1999.