PAPER *Special Section on Discrete Mathematics and Its Applications*

A Via Assignment and Global Routing Method for 2-Layer Ball Grid Array Packages

Yukiko KUBO[†a)] *and* **Atsushi TAKAHASHI**[††], *Members*

SUMMARY In this paper, we propose a global routing method for 2-layer BGA packages. In our routing model, the global routing for each net is uniquely determined by a via assignment of each net. Our global routing method starts from an initial monotonic via assignment and incrementally improves the via assignment to optimize the total wire length and the wire congestion. Experimental results show that our proposed method generates a better global routing efficiently.

key words: *ball grid array, monotonic routing, via assignment, wire congestion, total wire length, heuristic*

1. Introduction

In recent VLSIs, the number of required I/O pins becomes over several hundreds. Instead of DIP (dual in-line package) or QFP (quad flat package) in which the number of available I/O pins is small, BGA (ball grid array) package is used to realize the huge number of connections between VLSI and PCB. In BGA package, a connection between a bonding finger and a solder ball is realized using several routing layers. A bonding finger, which is called simply finger, is connected to VLSI chip by bonding wire. A solder ball, which is called simply ball, is connected to PCB.

Since the structure of a basic BGA package is symmetrical, a symmetrical radial routing can be applied if a netlist between fingers and balls is not specified. However, since a netlist between fingers and balls is usually given and often changed, and since a BGA package of various types is used, routing design is required repeatedly in design scene. Although there are several routing tools for VLSI package, the performance of these tools is not so good due to several features in package routing which are not in VLSI routing or in PCB routing. For example, the routing area of a BGA package has various huge obstacles which include balls themselves. The size of a ball is large compared to the width of a wire. For other example, a plating lead which realizes a connection to the outer ring surrounding package is sometimes required to every net in order to realize electric plating to all wires on surface. A plating lead is redundant for circuit operation but used to reduce the manufacturing cost since chemical plating is expensive.

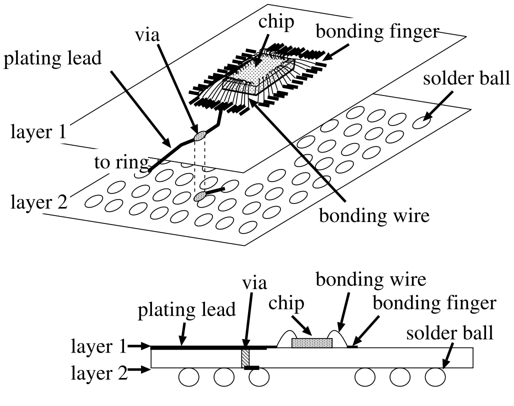

Fig. 1 BGA package and a route in it.

The first approach for BGA package routing was proposed in [1] and it was improved in [2]. In these approaches, it is assumed that the number of routing layers is one and that a netlist is not given. These approaches generate a netlist and generate a global route of each net. Each net connects a pair of a finger and a ball. The objective is to balance the congestion over the routing area and shorten the wire length of each net. Since a netlist is usually given in package design, these approaches are not used in package routing but used in package architecture design. For a given netlist in 2-layer BGA model as shown in Fig. 1, global routing on layer 1 may be possible by using these algorithms if a candidate of via positions is considered as a ball. But the feasibility of the global routes on layer 2 is not guaranteed.

While, the algorithms for multi-layer BGA and PGA (pin grid array) routing are proposed in [3] and [4], respectively. These algorithms first assign each net to a layer and then generate routes in each layer. However, the feasibility of routes from the finger of each net to the assigned layer and routes from the assigned layer to the ball of the net is not guaranteed. These routings require vias that are large compared to the wire width and almost same as ball interval in most cases. These algorithms skip the via assignment planning which is the most difficult part in package routing.

To optimize routes on both layer 1 and layer 2 by exchanging a via assignment, we propose a via assignment and global routing method for 2-layer BGA packages which considers total wire length and wire congestion. We introduce the concept of monotone in global routing and in via

Manuscript received August 23, 2004.
Manuscript revised November 19, 2004.
Final manuscript received January 11, 2005.

[†] The author is with the Department of Information and Media Sciences, The University of Kitakyushu, Kitakyushu-shi, 808-0135 Japan.

[††] The author is with the Department of Communications and Integrated Systems, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.

a) E-mail: kubo@env.kitakyu-u.ac.jp

DOI: 10.1093/ietfec/e88–a.5.1283

assignment. If a monotonic via assignment is given, the monotonic global routing on layer 1 is uniquely determined. Our method explores monotonic via assignments to reduce the total wire length and wire congestion of corresponding global routing. Our method starts from an initial monotonic via assignment and incrementally improves the via assignment. The proposed method was implemented by C++ and applied to some data. The experimental results show that our method efficiently generates better global routes with respect to total wire length and wire congestion.

2. Preliminaries

2.1 Problem Definition

In our BGA package model, the number of routing layers is two. Fingers, which are connected to a bare chip by bonding wire, are placed on the perimeter of a rectangle enclosing the chip on layer 1. Balls, which are connected to PCB, are placed on a grid array on layer 2 on the outside of the perimeter on which fingers are placed[†]. A net in BGA package consists of a finger and a ball. The connection between the finger and the ball of a net is realized by wires on layer 1 and 2 and vias that connect a wire on layer 1 and a wire on layer 2. Moreover, a ring structure which surrounds package is fabricated in our model. Each net should be connected to the ring in order to enable electrical plating to protect its wires. The extra connection to the ring of a net is called a plating lead. The ring is cut when the package is used. A plating lead is redundant for the operation, but it is used to reduce the fabrication cost and to improve the reliability.

Since the radius of a ball is large compared to the interval of the balls, the number of the routes between adjacent balls on layer 2 is limited. Therefore, the length of a wire connecting a ball and a via on layer 2 should be shortened and plating leads should be routed on layer 1. Therefore, the routing model is restricted as follows: the route of each net has only one via placed near the ball of the net, the wire on layer 1 connects the finger of the net and the ring through the via of the net, and the short wire on layer 2 connects the ball and the via of the net. The BGA package of our model and a routing example are shown in Fig. 1.

We assume that the number of vias to be placed in the area surrounded by adjacent four balls is at most 1. Therefore the set of possible positions of vias is represented by the set of nodes N_v on a grid array, whose interval is the same as that of balls as shown in Fig. 2. We call the grid *via grid array*. We partition the package area into four by cut lines as shown in Fig. 2 and apply our method to each part. In the following, we consider the bottom part.

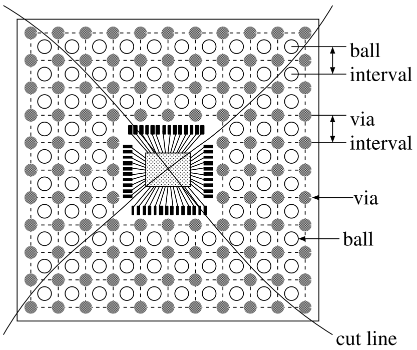

Fig. 2 Via grid array N_v.

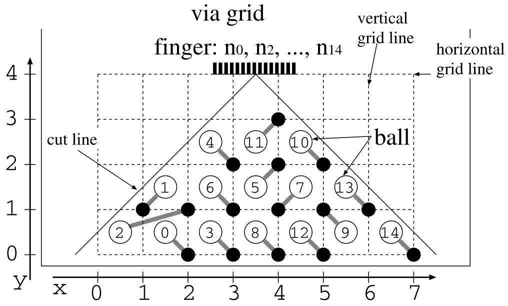

Fig. 3 Fingers and balls on a package, a via grid array based on the positions of balls, and a via assignment.

Nets are labeled according to the order of fingers on the perimeter from the left to the right as $n_0, n_1, \ldots, n_{m-1}$, where m is the number of nets. The ball and the via of net n_i are denoted by b_i and v_i, respectively. An assignment of the via of every net to a node in N_v is denoted by V. The coordinate of v_i in an assignment V is denoted by (x_i, y_i), where x_i and y_i are both integer. The coordinate of b_i is also denoted by the coordinate (x_i^b, y_i^b) using the axis of the via grid array, where x_i^b and y_i^b are always expressed by $\alpha + 0.5$ (α is integer). The x-coordinate of the left and right boundary at horizontal grid line $y = k$ are denoted by $x_{min}(k)$ and $x_{max}(k)$, respectively.

In Fig. 3, an example of a via assignment as well as the relative positions of fingers, balls, the via grid array, and cut lines are illustrated by taking all assumptions and definitions into account. In the example in Fig. 3, $x_{min}(k) = k - 0.5$ and $x_{max}(k) = x_{MAX} - k + 0.5$ where x_{MAX} is the maximum x-coordinate in the via grid array.

Global routing problem for 2-layer BGA package is defined in the following.

Global routing problem for 2-layer BGA

Inputs: Fingers and balls
Netlist (connection requirements of every pair of a finger and a ball)

Outputs: A via assignment V on N_v
The corresponding global routes

Objective: Minimize total wire length and
maximum wire congestion

[†] In case that some balls are placed inside the perimeter on which fingers are placed, the routing area is divided along the perimeter and routing for each area is performed individually. Therefore, it is assumed that balls are placed outside the perimeter as shown in Fig. 2 for easy explanation.

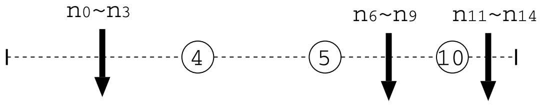

Fig. 4 Global routes of nets at $y = 2$.

2.2 Global Routing Based on a Via Assignment

In global routing problem for BGA packages, the outputs are a via assignment V and global routes corresponding to it.

If the route of each net on layer 1 intersects every horizontal grid line between its finger and the outer ring just only once, the route is said to be *monotonic*. Otherwise, it is said to be *non-monotonic*. It is easy to see that a monotonic routing is possible for via assignment V if and only if $x_i < x_j$ is satisfied for any pair of nets n_i and n_j $(i < j)$ such that $y_i = y_j$. A similar observation is found in [1]. A via assignment that satisfies this condition is said to be *monotonic*. In our method, only monotonic routes are dealt with in layer 1. Although a monotonic route tends to be shorter than a non-monotonic route, it may still become longer by snaking. The routes may violate a design rule if there are many routes between adjacent vias. Therefore a via assignment such that the total wire length and the maximum wire congestion of global routes are minimum should be pursued.

As for routes on layer 2, the route from a ball is expected to be short. But some routes may cross if no detour is allowed. In monotonic via assignment V, if $y_i = y_j$ for net n_i and n_j $(i < j)$, then $x_i < x_j$. Therefore, if $x_i^b > x_j^b$ and $y_i^b = y_j^b$ then routes from balls b_i and b_j on layer 2 cross if no detour is allowed. A via assignment V is said to be *detour-preventive* if $y_i \neq y_j$ for net n_i and n_j $(i < j)$ such that $x_i^b > x_j^b$ and $y_i^b = y_j^b$. This condition is considered while optimizing via assignments.

The via assignment shown in Fig. 3 is monotonic. Therefore, the monotonic global routing is uniquely determined. For example, three vias v_4, v_5 and v_{10} are assigned to the horizontal grid line $y = 2$. The route of a net in monotonic routing intersects the line as shown in Fig. 4. Similarly, the route of each net in monotonic routing intersects the other horizontal grid lines. The monotonic global route of each net on layer 1 is generated by connecting the intersecting point of every horizontal grid line. The global routing of the via assignment shown in Fig. 3 is shown in Fig. 5. In Fig. 5, the intersections of routes on each horizontal line are placed evenly between adjacent two vias. Obtained global routes are transformed into rubber band sketches [5]. If the rubber band sketches do not violate design rules specified by designers, physical routes can be obtained based on the sketches.

2.3 Evaluation of Global Routes

In the global routing problem, the objective is to minimize total wire length and maximum wire congestion. In this section, the way to evaluate wire length and wire congestion for a via assignment is discussed.

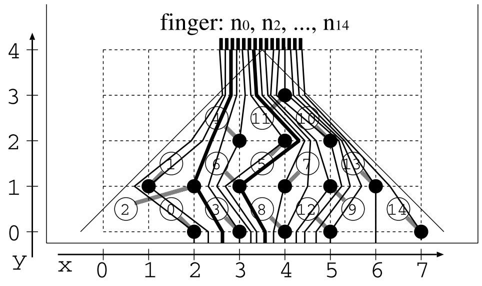

Fig. 5 Global routes of Fig. 3.

In via assignment V, the via assigned above v_i is denoted by $via(\text{above}, v_i)$. Formally, $via(\text{above}, v_i) = v_j$ if v_j is assigned above v_i on the same vertical grid line and no via is assigned between v_i and v_j on the line. If there is no such via, $via(\text{above}, v_i) = \emptyset$. Similarly, $via(\text{below}, v_i)$, $via(\text{left}, v_i)$, and $via(\text{right}, v_i)$ are defined.

2.3.1 Evaluation of Wire Length

If the wire length of a route on layer 1 becomes longer, the route snakes. If a route snakes, it crosses vertical grid lines. For example, routes of n_2 and n_6 in Fig. 5 (thick lines), have different length. The route of n_6 snakes and its wire length is long. As shown in Fig. 5, the route crosses vertical grid line $x = 4$ twice between horizontal lines $y = 1$ and $y = 3$. While, the route of n_2 does not snake and the route does not cross any vertical line. Therefore, we evaluate the wire length of a route on layer 1 by the number of intersections of the route and vertical grid lines. Although the exact wire length depends on the detailed routing, this evaluation has enough accuracy in global routing, since the distribution of wire lengths of detailed routings obtained according to the global routing is small.

For a monotonic via assignment V, the number of nets that intersect the vertical grid line between v_i and $via(\text{above}, v_i)$ is denoted by $cut_a(v_i)$ and defined as follows.

$$cut_a(v_i) = \begin{cases} |i-j|-1 & (\text{if } via(\text{above}, v_i) = v_j) \\ 0 & (\text{if } via(\text{above}, v_i) = \emptyset) \end{cases}$$

Similarly, the number of nets that intersect the vertical grid line between v_i and $via(\text{below}, v_i)$ is denoted by $cut_b(v_i)$.

$$cut_b(v_i) = \begin{cases} |i-k|-1 & (\text{if } via(\text{below}, v_i) = v_k) \\ 0 & (\text{if } via(\text{below}, v_i) = \emptyset) \end{cases}$$

Note that $cut_a(v_i) = cut_b(v_j)$ if $via(above, v_i)$ is v_j.

Although a route on layer 2 may detour even if a via assignment is detour-preventive, the route is expected to be short and seldom detour. Therefore, the wire length of a net on layer 2 is evaluated only by the Manhattan distance between the via v_i and the ball b_i of the net, which is denoted by $d(v_i)$ and defined as follows.

$$d(v_i) = |x_i^b - x_i| + |y_i^b - y_i|$$

2.3.2 Evaluation of Wire Congestion

Given monotonic via assignment V, the intersections of monotonic routes and horizontal grid lines are determined. Even if intersections are placed evenly between adjacent two vias, design rule violation may occur if the interval of the intersections is narrow. Therefore smallest possible wire congestion is better. Since the number of intersections (including vias) on each horizontal grid line is constant, the maximum wire congestion becomes small if the wire congestion of every part is balanced. In this section, we define wire congestion and the balance of the wire congestion.

In our method, wire congestion is evaluated by the reciprocal of the intersection interval between adjacent two vias. The wire congestion left of via v_i is denoted by $density_l(v_i)$ and defined as follows.

$$density_l(v_i) = \begin{cases} \frac{i-j}{x_i - x_j} & (\text{if } via(\text{left}, v_i) = v_j) \\ \frac{i+1}{x_i - x_{\min}(y_i)} & (\text{if } via(\text{left}, v_i) = \emptyset) \end{cases}$$

Similarly, the wire congestion right of the via v_i is denoted by $density_r(v_i)$ and defined as follows.

$$density_r(v_i) = \begin{cases} \frac{k-i}{x_k - x_i} & (\text{if } via(\text{right}, v_i) = v_k) \\ \frac{m-i}{x_{\max}(y_i) - x_i} & (\text{if } via(\text{right}, v_i) = \emptyset) \end{cases}$$

In our model, the difference of $density_l(v_i)$ and $density_r(v_i)$, denoted by $F(v_i)$, is defined to evaluate the balance of wire congestion.

$$F(v_i) = |density_l(v_i) - density_r(v_i)|$$

For example in Fig. 5, the number of intersections on the left of v_4 is 4 and the length between the left cut line and v_4 is 1.5. Then, the wire congestion between the left cut line and v_4 is 3.33. Similarly, the wire congestions between v_4 and v_5, v_5 and v_{10}, and v_{10} and the right cut line are 1, 5 and 10, respectively. $F(v_4) = |density_l(v_4) - density_r(v_4)| = |3.33 - 1| = 2.33$.

The routing cost for monotonic via assignment V, which is denoted by $COST(V)$, is defined by the following cost function including above evaluations:

$$COST(V) = \sum_{k=0}^{m-1} (\alpha cut_a(v_k) + \beta d(v_k) + \gamma F(v_k)),$$

where α, β and γ are coefficients.

3. Via Assignment Method

3.1 Outline of the Method

Our via assignment method is based on a heuristic approach. First, the method gives an initial monotonic detour-preventive via assignment. Then the via assignment is modified by sequential via movements so as to improve the routing cost as much as possible. Via movements are iterated until the routing cost cannot be improved any more.

The routing cost should be reduced as much as possible by a via movement. In our method, such a via movement is determined by selecting one that reduces the routing cost the most.

3.2 Initial Via Assignment

If balls are placed in monotonic order, an initial via assignment is given by placing vias as same as the ball placement. In practical BGAs, most balls are placed in monotonic order, but some balls may not. Therefore, we assume that a monotonic ball sequence is obtained from a given ball sequence by exchanging two adjacent balls on the same y-coordinate (the exchange is applied at most once for each ball). For example, the ball sequence on $y = 1.5$ is n_1, n_6, n_5, n_7, and n_{13} in Fig. 3. It is not monotonic but the monotonic one is obtained by exchanging the position of two balls of net n_5 and n_6. For such a ball sequence, it can be divided into two monotonic subsequences. For example, the ball sequence (n_1, n_6, n_5, n_7, n_{13}) is divided into two monotonic subsequences (n_1, n_6, n_7, n_{13}) and (n_5).

A monotonic and detour-preventive via assignment is obtained if these two subsequences for each ball sequence are assigned on the different horizontal grid lines. The detailed procedure to obtain an initial via assignment is defined as follows.

Step 1: For each ball sequence on the same y-coordinate, divide the sequence into two monotonic subsequences.
Step 2: Determine y-coordinates of vias. For two monotonic subsequences of the ball sequence on $y = k$, vias of a subsequence are put on $y = k - 0.5$ and the other on $y = k + 0.5$.
Step 3: For vias on the same y-coordinate, sort them in monotonic order and place them from the left.

For the example in Fig. 3, each step is executed as shown in Fig. 6. In this initial via assignment, the number of vias assigned on a same horizontal line may be over the number of the grid nodes on the line. In such a case, an overflow via is moved to a grid node without via on another grid line keeping the monotonic and detour-preventive conditions. We omit the detail in this paper.

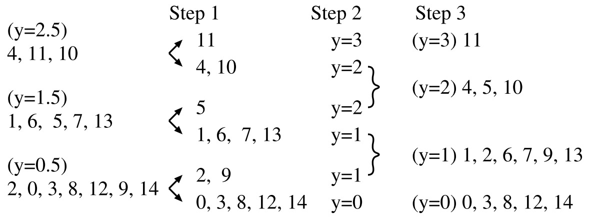

Fig. 6 Initial via assignment.

3.3 Sequential Via Movement

There are many ways to modify the current via assignment. In this paper, a sequential via movement is proposed for the via assignment modification. A sequential via movement is a movement in which vias are moved to their adjacent grid nodes one by one until reaching a grid node without a via as shown in Fig. 7.

Let $g(M)$ be the gain of a sequential via movement M which is defined as $COST(V) - COST(V')$, where V' is the via assignment obtained from V by applying M. Similarly, $\Delta cut_a(v)$ is defined as the difference between $cut_a(v)$ for V and $cut_a(v')$ for V', where v' in V' and v in V are assigned to the same grid node. $\Delta cut_b(v)$, $\Delta d(v)$, $\Delta density_l(v)$, $\Delta density_r(v)$, and $\Delta F(v)$ are defined similarly. In a sequential via movement M for V, the gain $g(M)$ can be represented as the sum of local gains $g_M(v)$ where v is a via contained in M. For example, we can define $g_M(v_6)$ for the sequential via movement M shown in Fig. 7 as

$$\begin{aligned} g_M(v_6) &= \Delta cut_b(v_6) + \Delta d(v_6) \\ &\quad + \Delta F(v_6) + \Delta F(v_2) + \Delta F(v_7) \\ &= (2-1) + (1-2) \\ &\quad + (|4-1| - |2-3|) + (|1-4| - |1-2|) \\ &\quad + (|1-2| - |3-2|) \\ &= 4 \end{aligned}$$

Around v_6, $cut_a(v_6)$ is changed by M, but $\Delta cut_a(v_6)$ is counted as $\Delta cut_b(v_4)$ in $g_M(v_4)$. Basically, the balance of wire congestion at a via is contained by the local gain of the via. However, $\Delta F(v_2)$ and $\Delta F(v_7)$ are also counted in $g_M(v_6)$ since $\Delta F(v_2)$ and $\Delta F(v_7)$ are changed by M but v_2 and v_7 are not contained by M. The via assignment V around v_6 and that obtained by applying M are shown in Fig. 8.

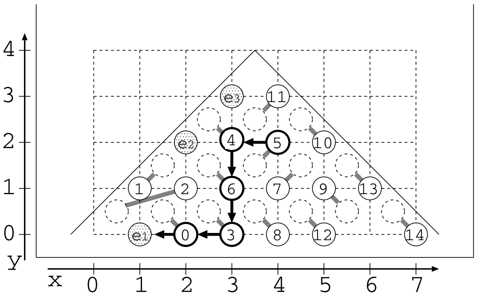

Fig. 7 A sequential via movement.

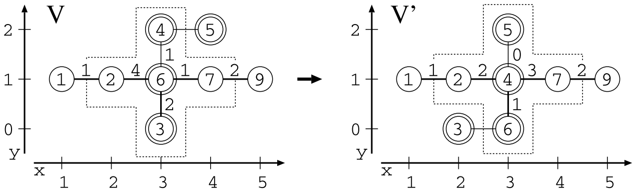

Fig. 8 The change of via assignment around v_6.

A sequential via movement is said to be monotonic if the direction of every horizontal movement of via is either left or right and that of every vertical movement is either above or below. In a monotonic sequential via movement M for V, $g_M(v)$ for a via v in M can be calculated if the previous two vias of v in M and the next via of v in M are given. For example, assume that a part of sequential via movement M is given as shown in Fig. 8. If M is monotonic, v_1, v_2, v_7 and v_9 remain the same grids after applying M, but may not if M is not monotonic. Therefore, $g_M(v_6)$ can be determined without using the other part of M if M is monotonic but can not otherwise.

3.4 Construction of Cost Graph

We construct a directed acyclic graph, which is called the cost graph of a via for via assignment V, that represents monotonic sequential via movements started from a via, which is denoted by v_s. In the cost graph G, the gain of a monotonic sequential via movement corresponds to the weight of a directed path in G. For our convenience, a dummy via is placed to a node if no via is assigned to the node in V. Therefore, a monotonic sequential via movement started from v_s is represented as $(v_s, v_i, v_j, v_k, \ldots, e)$ where e is a dummy via.

In a monotonic sequential via movement M for V, the local gain $g_M(v_k)$ can be defined as $g(v_i, v_j, v_k, v_c)$ where v_j is the previous via of v_k, v_i is the previous via of v_j, and v_c is the next via of v_k in M, since the local gain at v_k is same among all monotonic movements including sub-movements (v_i, v_j, v_k, v_c). v_c is $\emptyset$ if v_k is dummy.

Let G be the cost graph of via v_s for via assignment V. Since the type of a monotonic sequential via movement is either above-left, above-right, below-left, and below-right, G consists of four subgraphs corresponding to each type. The construction of the subgraph of the cost graph of v_s corresponding to below-left type is shown in the following.

Step 1: Generate node $(\emptyset, \emptyset, v_s)$

Step 2: Generate at most two nodes $(\emptyset, v_s, v_i)$ where v_i is $via(\text{below}, v_s)$ or $via(\text{left}, v_s)$ and v_i is allowed to be a dummy node.

Step 3: For each v_i in the below-left region of v_s, generate at most 4 nodes (v_i, v_j, v_k), where v_j is $via(\text{below}, v_i)$ or $via(\text{left}, v_i)$, and v_k is $via(\text{below}, v_j)$ or $via(\text{left}, v_j)$, where only v_k is allowed to be a dummy node.

Step 4: Add an edge from a node (v_i, v_j, v_k), to a node (v_j, v_k, v_c) with the weight:

$$\begin{cases} g(v_i, v_j, v_k, v_c) & \text{(if } v_c \text{ is not dummy)} \\ g(v_i, v_j, v_k, v_c) + g(v_j, v_k, v_c, \emptyset) & \\ & \text{(if } v_c \text{ is dummy)} \end{cases}$$

if the via assignment after the corresponding via movement is monotonic and detour-preventive.

The cost graph of above-left and below-left types from

v_5 is shown in Fig. 9, where dotted arrows represent the via movement that the via assignment after the corresponding via movement violates the monotonic and detour-preventive routing condition and shaded nodes represent unreachable nodes because of the violations.

It is clear that the length of a path is equal to the gain of the sequential via movement based on the path.

3.5 Via Assignment Method

In via assignment V, the local cost of a via is defined as the sum of costs defined around the via like the local gain of a movement. It is expected that the gain of a sequential via movement started from a via with the maximum local cost is large. Therefore, first, our proposed method constructs the cost graph of a via with the maximum local cost. Moreover, in order to reduce the computation time, the method only constructs two subgraphs of the cost graph which correspond to the direction of the via such that the local cost of the via is reduced more. Then the method determines a directed path with the maximum weight in the graph. If the weight is positive, the corresponding sequential via movement is applied. Otherwise, the method constructs the cost graph of a via with the 2nd maximum local cost, and repeats it until positive weight path is found.

The method is summarized as the following procedure.

Via assignment and global routing method

```
Step 1: Give an initial monotonic detour-preventive via
        assignment by the procedure in Sect. 3.2
Step 2: Sort all vias by their local costs in descending order
Step 3: $i = 1$
Step 4: Let $i$-th via be $v_s$ and construct a cost graph
Step 5: Search a longest path on the cost graph
Step 6: If the length of the longest path is more than 0,
        apply a via movement corresponding to the longest
        path, and go to Step 2
Step 7: $i = i + 1$
Step 8: If $i \leq m$, go to Step 4
Step 9: Generate global routes corresponding to the via
        assignment
```

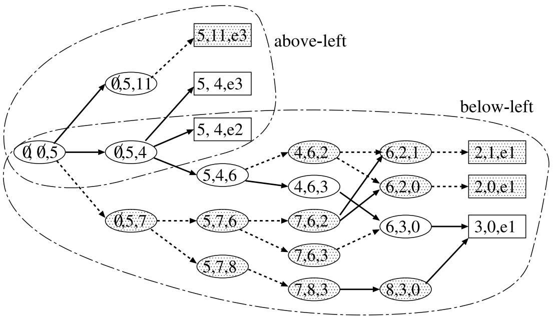

Fig. 9 Cost subgraph of v_5.

The number of nodes in the cost graph is $O(|N_v|)$ but it is proportional to m in practical because the package size is determined according to the net size. The longest path algorithm for a directed acyclic graph runs in $O(V^2)$, where V is the number of the nodes of a graph. In our method, the complexity of searching a longest path on the cost graph is $O(m)$ since the degree of each node is at most 3.

4. Experiments

We implemented our method by C++ language and applied it to some data, where all ball sequences in data1 and data2 are monotonic but some ball sequences in data3 and data4 are not. The program ran on a personal computer with CPU 450 MHz and 512 MB memory.

The coefficients α, β and γ of the cost function are all 1. Experimental results are shown in Table 1, where C, D, F, and $imp.$ are $\sum_{k=0}^{m-1} cut_a(v)$, $\sum_{k=0}^{m-1} \Delta d(v)$, $\sum_{k=0}^{m-1} \Delta F(v)$, and the percentage of (initial COST − output COST)/(initial COST), respectively. Each data is divided into four as shown in Fig. 2 and we applied our method to each divided part. The each value in Table 1 is the sum of the value of each part. The initial global routes and the output global routes of data3 are shown in Figs. 10(a) and (b), respectively. In Fig. 10(b), some crosses are found on layer 2 since a route on layer 2 is shown by a straight line segment. In order to remove these crosses, some detours are required on layer 2, but the increase of wire length expected to be small. It is clear that the proposed method can improve the total length of the routes and the wire congestion efficiently from experimental results.

5. Conclusion

In this paper, we proposed a via assignment and global routing method for 2-layer BGA packages. First we introduced a condition to determine global routes uniquely if a via assignment is given. Then our global routing method was proposed. The method is based on a heuristic algorithm. It generates an initial monotonic via assignment and modifies the via assignment to improve the total wire length and the

Table 1 Experimental results.

		initial				output					imp.	time
	# nets	C	D	F	COST	C	D	F	COST	# move	(%)	(s)
data1	316	175	316	356.45	847.45	82	350	215.87	647.87	73	23.6	5.54
data2	160	252	160	414.31	826.31	52	201	125.94	378.94	66	54.1	0.89
data3	160	250	168	434.74	852.74	93	219	146.55	458.55	103	46.2	2.38
data4	160	271	164	406.14	841.14	91	212	146.62	449.62	78	46.5	1.79

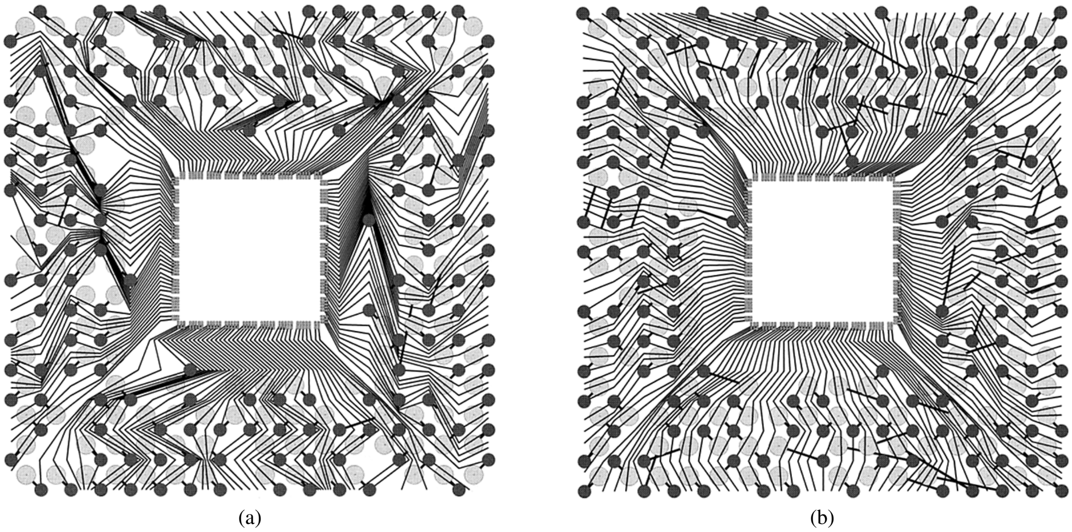

Fig. 10 The initial global routes (a) and the output global routes (b) by the via assignment method.

wire congestion by sequential via movements. The method iterates a sequential via movement until the cost is not improved any more. This method was programmed by C++ and applied to some data. The experimental results showed that the algorithm improved the total wire length and the wire congestion. To consider the global routes on layer 2 with no cross is one of our future works.

References

[1] M.-F. Yu and W.W.-M. Dai, "Single-layer fanout routing and routability analysis for ball grid arrays," Proc. Intl. Conf. Computer-Aided Design, pp.581–586, 1996.

[2] S, Shibata, K. Ukai, N. Togawa, M. Sato, and T. Ohtsuki, "A BGA package routing algorithm on sketch layout system," Journal of Japan Institute for Interconnecting and Packaging Electronic Circuits, vol.12, no.4, pp.241–246, 1997.

[3] 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 Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.17, no.2, pp.182–189, 1998.

[4] S.-S. Chen, J.-J. Chen, C.-C. Tsai, and S.-J. Chen, "An even wiring approach to the ball grid array package routing," Proc. Intl. Conf. Computer Design, pp.303–306, 1999.

[5] W.W.-M. Dai, R. Kong, J. Jue, and M. Sato, "Rubber band routing and dynamic data representation," Proc. Intl. Conf. Computer Design, pp.52–55, 1990.

Yukiko Kubo received B.E. degree in computer science and M.E. degree in electrical and electronic engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 1998 and 2000, respectively. She is currently a reasearch associate with Department of Information and Media Sciences, the University of Kitakyushu, Fukuoka, Japan. Her research interests are in VLSI layout design automation. She is a member of IEEE.

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 has been an associate professor since 1997. He visited University of California, Los Angeles, U.S.A., as a visiting scholar from 2001 to 2002. He is currently with Department of Communications and Integrated Systems, Graduate School of Science and Engineering, Tokyo Institute of Technology. His research interests are in VLSI layout design and combinational algorithms. He is a member of IEEE, and IPSJ.