

Routability Driven Via Assignment and Routing for 2-Layer Ball Grid Array Packages

Yoichi TOMIOKA[†] and Atsushi TAKAHASHI[†]

[†] Department of Communications and Integrated Systems, Tokyo Institute of Technology

E-mail: {yoichi, atsushi}@lab.ss.titech.ac.jp

Abstract Ball Grid Array packages in which I/O pins are arranged in a grid array pattern realize a number of connections between chips and a printed circuit board, but it takes much time in manual routing. We propose a fast routing method for 2-layer Ball Grid Array packages to support designers. Our method distributes wires evenly on top layer and increases completion ratio of nets by improving via assignment iteratively.

Key words ball grid array, monotonic, package, routing

1. Introduction

In current VLSI circuits, there can be hundreds of required I/O pins. Instead of Dual In-line Package (DIP) or Quad Flat Package (QFP), in which the number of available I/O pins is small, Ball Grid Array (BGA) packages are used to realize the huge number of connections between VLSI chips and printed circuit boards (PCBs).

Since the structure of a basic BGA package is symmetrical, symmetrical radial wiring can be applied if a netlist is not specified. In many instances, however, a netlist is given and this may change frequently though positions of most pads are usually fixed in advance. Because package design is processed concurrently with chip design and PCB design or follows them.

Auto routing on a package substrate is difficult because of critical design rule restrictions: package routing has several aspects that are not considered in VLSI routing or PCB routing. For example, the balls form huge obstacles in the routing area. The size of balls is large compared to the width of the wire. As another example, a plating lead, which realizes a connection to the outer ring surrounding the package, is sometimes required for every net in order to realize electric plating to all wires on the surface. Plating leads are redundant for circuit operation, but can reduce the manufacturing cost, because chemical plating is expensive. Therefore, still now, routing on a package substrate of complex model is often designed manually. However, by increasing the number of nets, package routing becomes a research target in order to decrease turn-around-time.

The first approach for BGA package routing was proposed in [1] and was improved in [2]. In these approaches, it is assumed that there is a single routing layer, and that a netlist is not given. These approaches generate a netlist and a global route for each net. Each net connects 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 routing design, these approaches are primarily used for package architecture design or flip-chip bonding design. For a given netlist in two-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. The feasibility of the global routes on layer 2, however, is not guaranteed.

Algorithms for multi-layer BGA and Pin Grid Array (PGA) routing have been proposed in [3] and [4], respectively. These algorithms first assign each net to a layer and then generate routes in each layer. However, neither the feasibility of the routes from the finger of each net to the assigned layer nor the routes from the assigned layer to the ball of the net are guaranteed. These routes require vias that are large compared to the wire width, and these algorithms omit the via assignment planning, which is the most difficult part of package routing.

A via assignment and global routing method for single-chip two-layer BGA packages that considers total wire length and wire congestion have been proposed as the first stage of package substrate routing in [5], and this method has been improved in [6]. In these papers, the concepts of monotonic global routing and monotonic via assignment focusing mainly on layer 1 are introduced. In the method, a via assignment is iteratively improved to minimize the maximum wire congestion on layer 1 while the total wire length on layer 2 is kept to be small enough.

Though the method achieves small total wire length and congestion, several enhancements are required in order to use the method in actual package routing design. In the method, since the via of a net is placed near the ball of the net, the wire of each net on layer 2 is short and the routing on layer 2 seems not to be difficult. However, there is no guarantee that 100% routing on layer 2 is possible. Moreover, in package substrate, various kinds of obstacles exist. For example, mold gates from which resin is poured into the package are placed on layer 1. In the region at which a mold gate placed, routing on layer 1 is not allowed but routing on layer 2 is allowed. Mold gates make it difficult to generate 100% routing since the via of a net may be placed away from its ball if the ball is under a mold gate. Even if the evaluation of a via assignment by cost function is better, it can not be adopted if 100% routing is impossible.

In this paper, we propose a via assignment and global routing method which is an enhancement of the method proposed in [6]. In our proposed method, the maximum wire congestion on layer 1 and the wire length of a net on layer 1 and

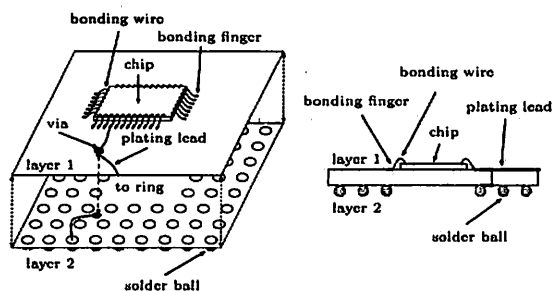


Fig. 1 A model of 2-layer BGA package

layer 2 are minimized. Moreover, a global routing on layer 2 is generated in the final stage to guarantee the 100% routing if a feasible via assignment is obtained. Our method consists of two phases. In the first phase, via assignment is iteratively improved to minimize the maximum wire congestion on layer 1 while the total wire length on layer 2 is kept to be small enough. Though this phase is based on the method proposed in [6], the computational complexity to obtain the maximum gain is improved from $O(N^2)$ to $O(N)$, where N is the number of grid nodes. In the second phase, via assignment is iteratively improved to improve the routability on layer 2 while the wire congestion on layer 1 is maintained. In this phase, global routing on layer 2 is generated. New modification to improve the routability on layer 2 while maintaining wire congestion on layer 1 is introduced.

2. Preliminary

2.1 Problem definition

In this paper, we consider a basic model of BGA package as shown in Fig. 1. Our BGA package model has two routing layers and single chip which is smaller than package size.

It has connection requirements between bounding fingers placed on the perimeter of a rectangle enclosing the chip on layer 1 and solder balls placed in a grid array pattern on layer 2. A bonding finger, which we will refer to as a finger, is connected to the chip by a bonding wire. 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 connection requirement between a finger and a ball is called a net, and is realized by wires on each layer and a via which connect a wire on layer 1 and a wire on layer 2. Moreover, there is a ring structure which surrounds package. Each net should be connected to the ring in order to enable electric 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 operation, but is normally used to reduce the fabrication cost and to improve the reliability.

A BGA package may have some mold gates in the corner of top layer to pour resin into the package. In the region at which a mold gate is placed, routing on layer 1 is not allowed but routing on layer 2 is allowed. The mold gates make it difficult to realize routes from vias to balls on layer 2 since the distance between vias and balls tend to be large.

The routing area of a package is usually divided into sectors. Our approach is applied to each sector. In the following, we focus on the bottom sector as shown in Fig. 2.

We assume that the number of vias to be placed in the area surrounded by four adjacent balls is at most 1. The set of candidate locations of vias which include the locations

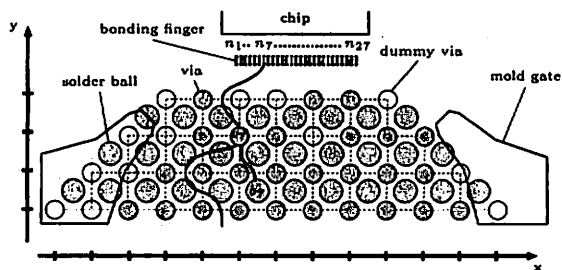


Fig. 2 Bottom sector

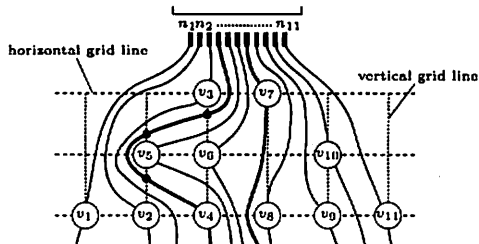


Fig. 3 A monotonic routing corresponding to a monotonic via assignment

within a mold gate is represented by the via array N . The interval of via array N is the same as that of the balls, and is unit length as shown in Fig. 2. An element in N is called a grid node.

In this paper, we assume that a net consists of a finger and a ball. Nets are labeled according to the order of fingers on the perimeter from the left to the right as n_1, n_2, n_3, \dots . Since the radius of a ball is large compared to the interval of the balls, the number of possible routes between adjacent balls on layer 2 is limited. Therefore, a via should be placed near its ball, and most of 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, the wire on layer 1 connects the finger of the net and the ring through the via of the net, and the wire on layer 2 connects the ball and the via of the net. The ball and the via of net n_i are denoted by b_i and v_i , respectively. The positions of b_i and v_i are denoted by (x_i^b, y_i^b) and (x_i^v, y_i^v) , respectively.

Let V be the set of vias of the nets. A via assignment to N is represented by bijection $\Phi: V \rightarrow N$, where E is the set of dummy vias.

The routing problem for a two-layer BGA package is defined as follows:

A routing problem for 2-layer BGA

Input: Fingers, balls, and netlist (Connection Requirements between fingers and balls)

Output: A via assignment Φ , corresponding routing on layer 1, and routing on layer 2

Objective: Minimize total wire length and maximum wire congestion

Constraint: All nets are realized, and vias are placed out of mold gates.

2.2 Monotonic via assignment

If the route of each net on layer 1 from its finger to the outer ring intersects every horizontal grid line only once, then

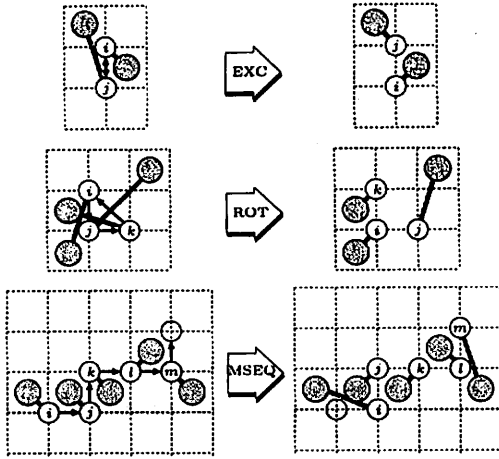


Fig. 4 Three modifications

the route is said to be monotonic. Otherwise, it is said to be non-monotonic. It is clear that a monotonic routing is possible for via assignment Φ if and only if $x_i^v < x_j^v$ is satisfied for any pair of nets n_i and n_j ($i < j$) such that $y_i^v = y_j^v$. A via assignment is said to be monotonic if a monotonic routing of layer 1 is possible [5], [6].

Given a monotonic via assignment, monotonic routing on layer 1 is uniquely determined. The via assignment shown in Fig. 3 is monotonic, and its routing is unique. For example, three vias v_5 , v_6 and v_{10} are assigned on middle row in Fig. 3. The route of nets n_1 , n_2 , n_3 , and n_4 in monotonic routing need to pass to the left of v_5 as shown in Fig. 3.

2.3 Evaluation of a via assignment

The wire length and wire congestion are used in the optimization of a via assignment as used in [6]. In this section, indices of a via assignment which are used in the evaluation of the via assignment is explained, briefly. Details are explained in [6].

The number of wires on layer 1 between v_i and the via above v_i is denoted by $cut_a(v_i)$. If no via exists above v_i , $cut_a(v_i)$ is zero. The Manhattan distance between via v_i and ball b_i of the net is denoted by $d(v_i)$.

The wire congestion of layer 1 between v_i and the via to the left of v_i is denoted by $density_l(v_i)$. That is, $density_l(v_i)$ is the number of wires of layer 1 between them over the distance between them. If no via exists to the left of v_i and v_i is in the routing region, then $density_l(v_i)$ is the wire congestion of layer 1 between v_i and the left boundary of routing region. If no via exists to the left of v_i and v_i is within the mold gates, then $density_l(v_i)$ is the number of wires of layer 1 which pass to the left of v_i over the distance between v_i and the left boundary of the sector. Similarly, $density_r(v_i)$ is defined.

The balance of wire congestion of v_i is denoted by $F(v_i)$. That is, $F(v_i) = |density_l(v_i) - density_r(v_i)|$.

2.4 Modifications

There are many ways to modify a via assignment. In [6], three simple modifications are proposed which are listed below.

(EXC) Two adjacent vias on a vertical grid line are exchanged

(ROT) Three vias on a unit square on the via grid array are rotated

(MSEQ) Vias are moved to their adjacent grid nodes on a via array one by one until reaching a grid node without a via in which the direction of every horizontal movement of vias is either left or right and that of every vertical movement is either above or below.

Examples of above three modifications is shown in Fig. 4.

3. Outline of our method

In our proposed method, the initial monotonic via assignment is generated by the method proposed in [5]. Then the initial via assignment is iteratively improved. Our method consists of two phases. In the first phase, via assignment is iteratively improved under the monotonic condition to minimize the maximum wire congestion on layer 1 while the total wire length on layer 2 is kept to be small enough. In the second phase, via assignment is iteratively improved under the monotonic condition to improve the routability on layer 2 while the wire congestion on layer 1 is maintained.

The first phase is based on the method proposed in [6]. In this phase, three types of modification {EXC, ROT, MSEQ} are used. In each iteration, a modification with the maximum gain on EXCs, ROTs, and MSEQs is applied to current via assignment to improve the wire congestion. Though the initial via assignment has vias placed on an obstacle, all vias are moved in feasible region in this iterative modification. In [6], it takes $O(|N|^2)$ time to find a MSEQ with the maximum gain. However, we show that it can be obtained in $O(|N|)$. In the first phase of our proposed method, each iteration takes only $O(|N|)$ time.

In the second phase, the via assignment generated by the first phase is iteratively modified by CEXC which is proposed here. In the second phase, the wire congestion on layer 1 is maintained, but the routability on layer 2 is improved. In each iteration, a CEXC which has the maximum gain among CEXCs which do not increase the maximum congestion is applied. The cost of second phase includes the number of unconnected nets. A routing graph corresponding to the routing problem on layer 2 is constructed, and routes are generated on it.

Note that in each iteration, a modification by which a monotonic via assignment becomes non-monotonic is not applied since via assignments are restricted to be monotonic.

4. The first phase

4.1 Cost of a via assignment

The routing cost for monotonic via assignment used in [6] is extended to move vias out of mold gate since the initial via assignment may have vias which are on a mold gate.

If a via $v \in V$ is on a mold gate, then $obs(v) = 1$. Otherwise $obs(v) = 0$.

The routing cost for monotonic via assignment Φ , which is denoted by $COST(\Phi)$, is defined as follows:

$$COST(\Phi) = \sum_{v \in V} (\alpha_1 cut_a(v) + \beta_1 d(v) + \gamma_1 F(v) + \delta_1 obs(v))$$

where α_1 , β_1 , γ_1 , and δ_1 are coefficients.

Note that δ_1 is set to much larger than the others in order to obtain a feasible via assignment.

4.2 The computational complexity

In our algorithm, a modification with the maximum gain among all EXCs, ROTs, and MSEQs under the monotonic condition is applied in each iteration.

In our method, a modification with the maximum gain un-

der the monotonic condition is selected and applied to the current via assignment. For EXC and ROT, the number of patterns is $O(|N|)$, which is small enough to enumerate all the patterns.

In order to find a MSEQ with the maximum gain the cost graphs are constructed. A cost graph is a directed acyclic graph (DAG) such that every directed path from a source to a sink corresponds to a MSEQ and the length of the path corresponds to the gain by the MSEQ. A MSEQ with the maximum gain is obtained by generating cost graphs and searching a longest path on the graphs.

In [6], the cost graph for every MSEQ beginning with a via is constructed and a longest path in each cost graph is obtained. A longest path of DAG can be obtained in $O(n+m)$, where n and m are the number of vertices and edges, respectively. Since the numbers of vertices and edges in a cost graph for each via are $O(|N|)$, we can obtain a longest path in $O(|N|)$ for each cost graph. The maximum gain on MSEQs can be obtained in $O(|N|^2)$ since the number of cost graphs is $O(|N|)$.

In the next section, we show that the cost graphs beginning with different via can be combined. Since just four cost graphs are constructed where the numbers of vertices and edges are $O(|N|)$, a MSEQ with maximum gain can be obtained in $O(|N|)$.

4.3 Improved cost graph

The type of an MSEQ is either above-left, above-right, below-left, or below-right. One cost graph is constructed for each type. Though four cost graphs are constructed for each via in [5] and [6], the number of cost graphs which is constructed in our method is just four.

An MSEQ as shown in Fig. 4 is represented by a sequence of vias as $(v_i, v_j, v_k, v_l, v_m, e)$, where e is a dummy via.

Let $g(M)$ be the gain of an MSEQ M that is defined as $COST(\Phi) - COST(\Phi')$, where Φ' is the via assignment obtained from Φ by applying M . For any MSEQ, $g(M)$ can be represented as the sum of local gains $g_M(v_i)$, where v_i is a via contained in M . Then, $g_M(v_i)$ can be calculated as described in [5]. The local gain $g_M(v_i)$ can be defined as $g(v_{pp}, v_p, v_i, v_n)$ where v_n is the next via of v_i in M , v_p is the previous via of v_i in M , and v_{pp} is the previous via of v_p in M . In other words, the local cost $g_M(v)$ can be obtained if the subsequence of M which consists of four vias around v_i is known. Even if two MSEQs M_1 and M_2 begin with different vias, $g_{M_1}(v)$ and $g_{M_2}(v)$ are the same if the subsequences of M_1 and M_2 around v_i are the same.

In a cost graph, a subsequence (v_{pp}, v_p, v_i, v_n) corresponds to vertex (v_{pp}, v_p, v_i) , vertex (v_p, v_i, v_n) , and the edge between them with weight $g_M(v_i)$. Each vertex of a cost graph is labeled by the sequence of three vias. There is a one-to-one correspondence between MSEQs and paths of the cost graph, and the gain of an MSEQ is the length of the corresponding path.

Since the type of an MSEQ is either above-left, above-right, below-left, or below-right, the cost graph consists of four subgraphs corresponding to each type. The algorithm of the above-right type cost graph construction is shown in Fig. 5. In a cost graph, the number of vertices to which a label in which v_i is the last element is assigned is at most seven. The number of edges incident from a vertex is at most two. Therefore, the numbers of vertices and edges of a cost graph are $O(|N|)$. For example, in the via assignment shown in Fig. 6, labels in which v_i is the last element are (LL, L, v_i) , (BL, L, v_i) , (BL, B, v_i) , (BB, B, v_i) , (\emptyset, L, v_i) , (\emptyset, B, v_i) , and

```

ConstructCostGraph(A via assignment  $\Phi$ )
 $X \leftarrow V \cup E$ 
 $C \leftarrow$  the set of the most above-right vias in  $X$ 
while  $C \neq \emptyset$  do
  select  $v$  from  $C$ 
  if  $v$  is dummy then
    generate feasible vertices  $(\emptyset, v_p, v)$  and  $(v_{pp}, v_p, v)$ 
  else
    if  $(\emptyset, v, v_n)$  exists then
      generate  $(\emptyset, \emptyset, v)$ 
      generate an edge from  $(\emptyset, \emptyset, v)$  to  $(\emptyset, v, v_n)$ 
    if  $(v_p, v, v_n)$  exists then
      generate feasible vertices  $(\emptyset, v_p, v)$  and  $(v_{pp}, v_p, v)$ 
      generate edges from these vertices to  $(v_p, v, v_n)$ 
   $X \leftarrow X \setminus \{v\}$ 
   $C \leftarrow$  the set of the most above-right vias in  $X$ 
done

```

Fig. 5 Algorithm of graph construction for above-right direction

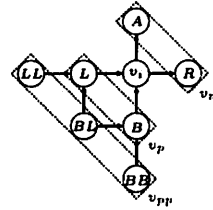


Fig. 6 Vias used to calculate $g_M(v_i)$

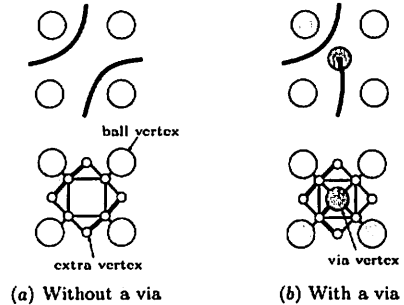


Fig. 7 Routing subgraphs on layer 2

$(\emptyset, \emptyset, v_i)$. The edges incident from (LL, L, v_i) are (L, v_i, A) and (L, v_i, B) .

Note that we do not generate a vertex if a via assignment becomes non-monotonic when the modification corresponding to the vertex is applied.

5. The second phase

In the first phase, a via assignment where wires are distributed evenly is obtained. So, in the second phase, the via assignment is modified to improve the routability on layer 2 while the routing quality on layer 1 is maintained. In the second phase, we use another cost defined in section 5.3, since the target is different from the first phase.

5.1 Routing graph on layer 2

A routing graph representing routing resource on layer 2 is constructed. The routing graph has ball vertices, via vertices, and extra vertices. A ball vertex and a via vertex correspond to a ball and a via, respectively. The number of routes intersecting between two adjacent balls is at most one since ball radius is so big. A subgraph of a routing graph in

Fig. 7(a) corresponds to a grid in which a ball exists in each corner and to which a via is not assigned. A subgraph of a routing graph in Fig. 7(b) corresponds to a grid to which a via is assigned.

A global routing on layer 2 is obtained by using a rip-up and reroute technique on the routing graph. A shortest path of each net is sequentially generated on the routing graph regarding the routes of the other nets as obstacles. If the route of a net can not be found, then a shortest path without obstacles by other nets is generated, and the routes of the other net which intersect the found shortest path are ripped up. Whenever a route is ripped up, the weight of each vertex on the ripped up route and on the found shortest path is increased to avoid iterations such as the routes of two nets are alternately generated and ripped up.

5.2 Local modification for routability

Though a via is placed near its ball in the output of the first phase, any net can not be connected if the via assignment is bad. So, in the second phase, the via assignment generated by the first phase is iteratively modified to improve the routability while the wire congestion on layer 1 is maintained.

In the first phase, wire congestion is minimized effectively by using EXCs, ROTs, and MSEQs since these modifications change routes on layer 1 drastically. However, in the second phase, routes on layer 1 should not be changed drastically to maintain the quality. So, we propose a local modification CEXC which exchanges vias v_i and v_j . By restricting a pair of vias, a local modification CEXC has high probability improving the routability on layer 2 though the effect on routes on layer 1 is small.

Even if v_i and v_j are exchanged under the monotonic condition, routes on layer 1 are changed only around v_i and v_j . If the difference of i and j is small, then the number of routes changed is also small. Therefore, the vias which can be exchanged to v_i are restricted in CEXCs.

If v_i is not dummy via, then v_i can be exchanged to v_j only if $i - 2 \leq j \leq i + 2$. If v_i is dummy via, then v_i can be exchanged to the via of a net only if the route of the net passes near v_i . Formally, such nets are defined as follows: Let v_l and v_r be vias to the left and right of v_i , respectively. A net whose route passes near v_i is n_j ($k - 2 \leq j \leq k + 2$), where

$$k = \frac{l \cdot (x_r^v - x_i^v) + r \cdot (x_i^v - x_l^v)}{x_r^v - x_l^v}.$$

This modification is called CEXC.

In addition, CEXC is restricted not to increase the maximum wire congestion on layer 1, and to satisfy the monotonic condition after it is applied. Let c and c' be the maximum congestion around v_i and v_j before they are exchanged and after they are exchanged, respectively. Let MAX be the maximum congestion allowable congestion of layer 1 to satisfy the design rule. If $c \leq MAX$, then CEXC is feasible only if $c' \leq MAX$. Otherwise, CEXC is feasible only if $c' < c$.

EXC may drastically change routes of layer 1 while the distance between the via and the ball is maintained, whereas CEXC may improve the routability of layer 2 while the maximum wire congestion of layer 1 is maintained.

5.3 Evaluation of a via assignment

In the second phase, the routes of layer 2 are generated in order to improve the routability of layer 2.

Let U be the number of unconnected nets on the routing graph. The cost of the second phase includes U to obtain a

via assignment such that all nets are realized.

In addition, the cost includes the estimated total wire length on layer 2, since the routability on layer 2 is improved if the total wire length on layer 2 decrease. Let L be the total wire length on layer 2.

Though the output of the first phase distributes wires evenly, the maximum congestion of layer 1 may violate the design rule. However, the design rule may be satisfied by adjusting via positions within a square surrounded by four balls if the violation of wire congestion is small. So, the amount of violations should decrease, and the cost is added to estimate the violation.

If $density_i(v_i) \leq MAX$, then the violation between v_i and the via to the left of v_i is 0. Otherwise, the violation is $density_i(v_i) - MAX$. Let Δ be the sum of the violations for a whole via grid. In the second phase, the cost of a via assignment is defined as follows:

$$COST(\Phi) = \alpha_2 L + \beta_2 \Delta + \gamma_2 U$$

where α_2, β_2 , and γ_2 are coefficients.

Note that γ_2 is set to much larger than the others in order to connect more nets.

Even if CEXC hardly change routes on layer 1, the wire length on layer 1 may increase in the second phase. However, even if the evaluation of a via assignment by cost function is better, it can not be adopted if 100% routing is impossible or the maximum wire congestion does not satisfy the design rule. Therefore, the routability has priority over the wire length on layer 1 in the second phase.

6. Experiments and Results

We implemented the proposed method in C++ language and applied it to a number of test cases which has mold gates as shown in Fig. 2. The program ran on a personal computer with a 3.4GHz CPU and 1 GB of memory.

In the first phase, execution times between the method proposed in [6] and our proposed method are compared. It is expected that our proposed method can obtain the same via assignment faster than the method in [6] since we improved the computational complexity to obtain a modification with the maximum gain from $O(|N|^2)$ to $O(|N|)$. In the second phase, we show that enough routability can be obtained in most of the test cases by CEXC.

In our experiment, the total wire length of routing on layer 2 is the number of used edges on routing graph. $\alpha_1, \beta_1, \gamma_1$, and β_2 are set to 1. δ_1 and γ_2 are set to much larger than the others. α_2 is set to $\frac{1}{4}$. We assume that MAX is $h + 1$ where h is the number of rows of balls, and $MAX = 5$ in all of data since the number of rows is 4.

Table 1 shows the result of the first phase. C, D, and F are $\sum cut_a(v_i)$, $\sum d(v_i)$, and $\sum F(v_i)$, respectively, and COST is the sum of C, D, and F. OLD is execution time of the method proposed in [6], and PROP. is execution time of our proposed method where the cost graph is improved.

The output of the first phase for most inputs is improved drastically as shown in Table 1. Although the method proposed in [6] needs 45 second for data1, our proposed method obtains the identical output within 3 second due to improving the computational complexity of each iteration.

Table 2 shows the result of the second phase. AVE. LENG. is the average of the wire length on layer 2, and ERR. is the number of unconnected nets for routing on layer 2.

In this experiment, all nets are realized in data2, data3, and data4, and AVE. LENG. gets lower than the output of

Table. 1 The result in the first phase

	#net	Cost								#Mod				Time [s]		
		Initial				First Phase				EXC	ROT	MSEQ	ALL	OLD	PROP.	
		C	D	F	COST	C	D	F	COST							IMP. [%]
data1	316	255	316	398.72	989.72	118	412	233.98	763.98	21.2	11	0	48	59	44.96	2.58
data2	192	447	192	706.00	1345.0	147	284	349.67	780.67	41.9	20	5	32	57	8.37	0.86
data3	160	307	168	503.55	978.55	81	255	165.52	501.52	48.7	16	5	57	78	10.58	1.29
data4	160	353	164	471.42	988.42	63	241	171.40	475.40	51.9	21	2	45	68	9.57	1.09

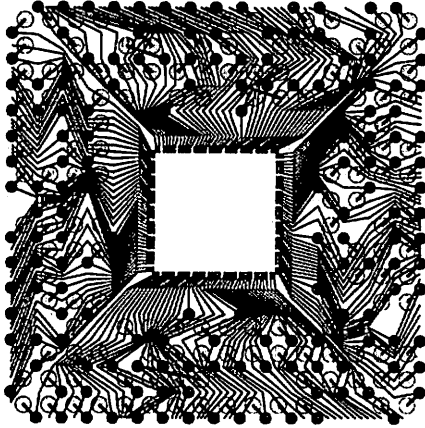


Fig. 8 The initial routes on layer 1 for data4

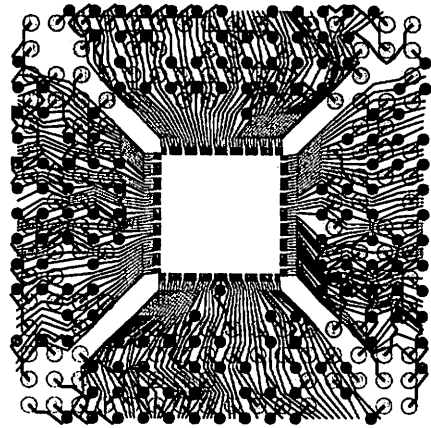


Fig. 9 The output routes on layer 1 and layer 2 for data4

Table. 2 The result in the second phase

	First Phase		Second Phase			
	AVE. LENG.	ERR.	AVE. LENG.	ERR.	#Mod	Time [s]
data1	2.90	11	2.87	3	20	61.32
data2	3.75	3	3.43	0	12	3.33
data3	4.21	5	3.76	0	21	4.31
data4	3.78	3	3.59	0	9	2.31

the first phase.

The initial routes for data4 is shown in Fig. 8, and the output of the second phase is shown in Fig. 9. Mold gates are not drawn in these figures. Our proposed method explores monotonic via assignments effectively, a via assignment which guarantees 100% routability on layer 2 is obtained in many test cases.

On the other hand, our method does not realize all net in data1 where the output of the first phase has many unconnected nets. In addition, though most wires are distributed evenly, there exist places with high wire congestion near mold gates. This is caused by that moving vias out of mold gates has priority over improving or maintaining the wire congestion and the distance between a via and a ball in the first phase. These troubles will be improved if the initial via assignment is created out of a mold gate with the routability analysis, and the wire congestion on layer 1 can be improved if parts of plating leads is realized on layer 2. But, these are in our future works.

7. Conclusion

We showed that a modification with the maximum gain is obtained in $O(|N|)$, though it takes $O(|N|^2)$ times in [6]. Moreover, we gave a routing graph for routing on layer 2, and a local modification to improve the routability on layer 2 while the maximum wire congestion is maintained. In our experiments, our method obtains a via assignment which distributes wires evenly faster than the method proposed in [6],

and the routability on layer 2 is improved drastically. Our proposed method explores monotonic via assignments effectively, a via assignment which guarantees 100% routability on layer 2 is obtained in many test cases.

In our future work, we will propose the method such that the initial via assignment is created out of a mold gate with the routability analysis. Moreover, we will consider the method where parts of plating leads are realized on layer 2.

References

- [1] M.F. Yu and W.W.M. Dai, "Single-Layer Fanout Routing and Routability Analysis for Ball Grid Arrays," Proceedings of International Conference Computer-Aided Design, pp.581-586, 1995.
- [2] 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).
- [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 Transactions on Computer-Aided Design of Integrated Circuits and Systems, 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," Proceedings of International Conference on Computer Design, pp.303-306, 1999.
- [5] Y. Kubo and A. Takahashi, "A Via Assignment and Global Routing Method for 2-Layer Ball Grid Array Packages," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol.E88-A, no.5, pp.1283-1289, 2005.
- [6] Y. Kubo and A. Takahashi, "Global Routing by Iterative Improvements for 2-Layer Ball Grid Array Packages," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD), vol.25, no.4, pp.725-733, 2006.