Feasible Region Assignment of Routing Nets in Single-Layer Routing

Jin-Tai Yan[1], Yu-Jen Tseng[1] and Chia-Heng Yen[2]
[1]Department of Computer Science and Information Engineering, Chung-Hua University
Hsinchu, Taiwan, R.O.C
[2]Department of Computer Science and Engineering, National Taiwan Ocean University
Keelung, Taiwan, R.O.C

Abstract –It is well known that single-layer routing is used for RDL routing in flip-chip designs and substrate routing in package designs. In this paper, given a set of two-terminal nets in a single-layer gridded routing plane, the routing regions of all the given nets can be initially constructed. Based on the routing constraints on *different intersection conditions of two routing regions in a single layer, the wiring directions of the given nets can be further assigned. Finally, based on the assigned directions of the given nets, the wiring paths of the given nets onto the routing grids can be assigned by diffusing the overlapping paths and eliminating the unnecessary detours in single-layer routing. The experimental results show that our proposed approach can route 99.98% of the given nets in single-layer routing for 6 tested examples in reasonable CPU time on the average.*

I. INTRODUCTION

As the complexity increases and the feature size decreases in modern VLSI designs, the requirement of more IO pads becomes a critical issue. Recently, an IC package uses a ball grid array(BGA) substrate to connect a die with the wire-bonding or flip-chip interconnections to the substrate. However, the high-density integration in an IC package makes the off-chip interconnections difficult. For a wire-bonding die, the IO pads are connected to the corresponding bond pads through a set of bounding wires. On the other hand, an advanced packaging technology, *flip-chip(FC) package*[1], is introduced to meet the requirement of the higher integration density and the larger IO count in complicated VLSI circuits. For a flip-chip die, an extra metal layer, *redistribution layer(RDL)*[2-4], is firstly used to redistribute the IO pads to the bump balls without changing the locations of the placed IO pads. Furthermore, escape routing[5-6] breaks out all the bump pads to the chip boundary at the break pads in the given layers. Finally, a set of substrate interconnections from the bond pads in the wire-bonding dies or the break pads in the flip-chip dies to the solder balls in a BGA package must be connected in substrate layers.

For substrate interconnections, the routing concept was published in some related packages[7-8]. However, it is assumed that the bond pads or break pads in the published works are located side by side with respect to solder balls. Recently, Liu et al.[9-10] also developed an efficient yet effective algorithm for substrate topological routing. In the work, they explain why planar routing is still required with multiple routing layers for substrate routing and propose a flexible via-staggering technique to improve routability. Due to the limit of planar routing for substrate interconnections, single-layer routing becomes an important issue in an IC package.

In this paper, given a set of two-terminal nets in a single-layer gridded routing plane, the routing regions of all the given nets can be initially constructed. Based on the routing constraints on different intersection conditions of two routing regions in a single layer, the wiring directions of the given nets can be further assigned. Finally, based on the assigned directions of the given nets, the wiring paths of the given nets onto the routing grids can be assigned by diffusing the overlapping paths and eliminating the unnecessary detours in single-layer routing. The experimental results show that our proposed approach can route 99.98% of the given nets in single-layer routing for 6 tested examples in reasonable CPU time on the average.

II. PROBLEM FORMULATION

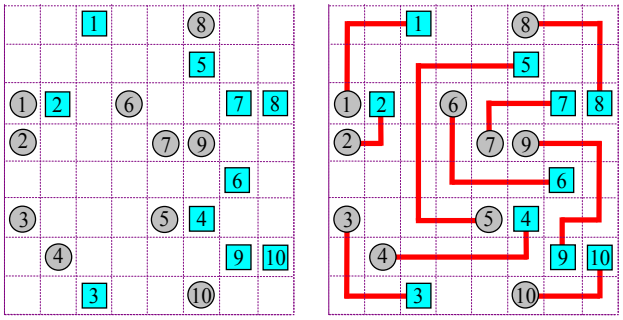
In general, single-layer routing is only allowed to route all the nets in a single routing layer. Hence, the *wire crossing* between any pair of nets cannot be allowed in the routing layer. Clearly, the *non-crossing constraints* in the given nets must be maintained in single-layer routing.

Given a set of n two-terminal nets, $N=\{N_1, N_2, ..., N_n\}$, in a single-layer gridded routing plane, it is assumed that $S=\{s_1, s_2, ..., s_n\}$ is a set of n start terminals and $T=\{t_1, t_2, ..., t_n\}$ is a set of n target terminals in N. Basically, any start terminal, s_i, $1 \le i \le n$, is connected to its corresponding target terminal, t_i, to form a two-terminal net, N_i, and the wirelength of the net, N_i, is defined as the number of the passing grids from the start terminal, s_i, to the target terminal, t_i. In single-layer grid-based routing model, it is known that any routing wire must be assigned onto a grid. Clearly, the single-layer routing problem in a grid-based routing plane is to assign the routing wires of the given nets onto feasible routing grids to maximize the number of routed nets with the minimization of the total wirelength in a single routing layer with satisfying the non-crossing constraint.

In single-layer routing, a given set of 10 two-terminal nets with the same net number on the start terminals labeled by circles and the target terminals labeled by rectangles are considered in an array of 8x8 routing grids as illustrated in Fig. 1(a). Furthermore, by distributing the feasible non-

978-1-4799-3432-4/14/$31.00 ©2014 IEEE

crossing routing wires of the given 10 nets onto the grid-based region, the 10 given nets can be completely routed in a single routing layer and the total wirelength of the 10 given nets can be minimized as 55 as illustrated in Fig. 1(b).



(a) (b)

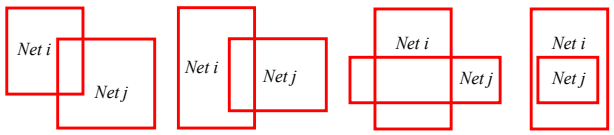
Fig. 1 Single-layer routing results of 10 given nets

III. Feasible Region Assignment of Routing Nets in Single-Layer Routing

Given a set of n two-terminal nets, $N=\{N_1, N_2, ..., N_n\}$, in a single-layer gridded routing plane, the region assignment of the given nets for single-layer routing can be divided into two sequential phases: *Determination of wiring direction* and *Region assignment for paths of routing net*.

3.1 Determination of Wiring Direction

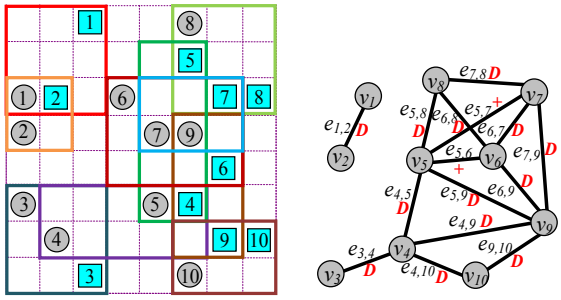
For any two-terminal net in Manhattan routing model, its *routing region* can be defined as the minimum rectangle covering the two given terminals. If the routing regions of two independent nets intersect, the final wiring paths of the routing nets may cross in single-layer routing. To route successfully all the given nets in a single layer, the different intersection conditions of two routing regions must be formally considered in single-layer routing. From geometrical viewpoint, for any pair of two routing nets, i and j, the intersection condition of two routing regions can be divided into four different intersections: *Diagonal intersection(D-intersection)*, *Adjacent intersection(A-intersection)*, *Crossing intersection(+-intersection)* and *Covering intersection(O-intersection)* as illustrated in Fig. 2.



(a) *D-intersection* (b) *A-intersection* (c) *+-intersection* (d) *O-intersection*

Fig. 2 Intersection conditions of two routing regions

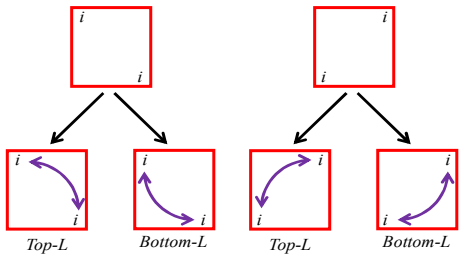
Given a set of two-terminal nets in a single layer, based on four different intersection relations of two routing regions for any pair of nets, an undirected intersection graph, $G(V, E)$, can be constructed as follows: each vertex, v_i, in V represents an routing net, i, and each edge, $e_{i,j}$, in E represents an intersection relation of two routing regions for nets, i and j. Refer to the routing specification in single-layer routing in Fig. 1(a), the routing regions of the given 10 nets, $N=\{N_1, N_2, N_3, N_4, N_5, N_6, N_7, N_8, N_9, N_{10}\}$, can be illustrated in Fig. 3(a). Furthermore, according to the intersection relation between two routing region, its corresponding intersection graph, $G(V, E)$, can be constructed as shown in Fig. 3(b), where $V=\{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}\}$ *and* $E=\{e_{1,2}, e_{3,4}, e_{4,5}, e_{4,9}, e_{4,10}, e_{5,6}, e_{5,7}, e_{5,8}, e_{5,9}, e_{6,7}, e_{6,9}, e_{7,8}, e_{7,9}, e_{9,10}\}$. Clearly, the 13 edges, $e_{1,2}$, $e_{3,4}$, $e_{4,5}$, $e_{4,9}$, $e_{4,10}$, $e_{5,8}$, $e_{5,9}$, $e_{6,7}$, $e_{6,8}$, $e_{6,9}$, $e_{7,8}$, $e_{7,9}$ and $e_{9,10}$, belong to *D-intersection* relations and the other 2 edges, $e_{5,6}$ and $e_{5,7}$, belong to *+-intersection* relations.



(a) (b)

Fig. 3 Routing regions of 10 given nets and its corresponding intersection graph

For any net, i, it is assumed that the wiring path inside its routing region must have no detour and the wiring shape may be a *top-L* or *bottom-L* direction as illustrated in Fig. 4.

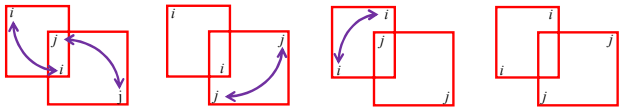


(a) (b)

Fig. 4 Two wiring directions of a routing net

For a diagonal intersection of two routing regions for nets, i and j, in Fig. 2(a), there are four intersection conditions according to the terminal locations of the two nets and the four intersection conditions are classified into three different types as illustrated in Fig. 5. To route successfully the nets, i and j, in a single layer, the wiring shape of the net, i or j, must be further constrained as a fixed wiring direction as follows: As the *D1-type* diagonal intersection in Fig. 5(a), the wiring shapes of the nets, i and j, must be constrained as a *bottom-L* direction and a *top-L* direction, respectively. As the *D2-type* diagonal intersection in Fig. 5(b), the wiring shape of the net, i, can be assigned as a *top-L* or *bottom-L* direction and the wiring shape of the net, j, must be constrained as a *bottom-L* direction. As the D2-type diagonal intersection in Fig. 5(c),

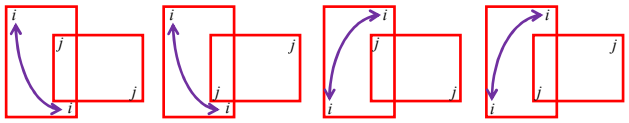
the wiring shape of the net, *i*, must be constrained as a *top-L* direction and the wiring shape of the net, *j*, can be assigned as a *top-L* or *bottom-L* direction. As the *D3-type* diagonal intersection in Fig. 5(d), the wiring shapes of the nets, *i* and *j*, cannot be simultaneously assigned as a *bottom-L* direction and a *top-L* direction, respectively.



(a) *D1-type* (b) *D2-type* (c) *D2-type* (d) *D3-type*

Fig. 5 Constraint of wiring directions on four diagonal intersections

For an adjacent intersection of two routing regions for nets, *i* and *j*, in Fig. 2(b), there are also four intersection conditions according to the terminal locations of the two nets and the four intersection conditions are classified into two different types as illustrated in Fig. 6. To route successfully the nets, *i* and *j*, in a single layer, the wiring shape of the net, *i*, must be further constrained as a fixed wiring direction as follows: As the *A1-type* adjacent intersection in Fig. 6(a) and 6(b), the wiring shape of the net, *i*, must be constrained as a *bottom-L* direction and the wiring shape of the net, *j*, can be assigned as a *top-L* or *bottom-L* direction. As the *A2-type* adjacent intersection in Fig. 6(c) and 6(d), the wiring shape of the net, *i*, must be constrained as a *top-L* direction and the wiring shape of the net, *j*, can be assigned as a *top-L* or *bottom-L* direction.



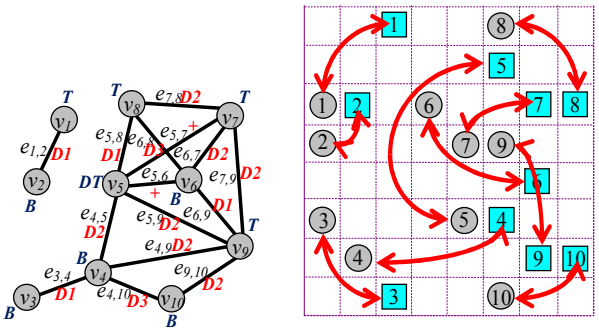
(a) *A1-type* (b) *A1-type* (c) *A2-type* (d) *A2-type*

Fig. 6 Constraint of wiring directions on four adjacent intersections

Besides that, for a crossing or covering intersection of two routing regions for nets, *i* and *j*, in Fig. 2(c) and 2(d), there are also four different intersection conditions according to the terminal locations of the two nets. However, there is no constraint in the assignment of the wiring shapes for the nets, *i* and *j*. To route successfully the nets, *i* and *j*, in a crossing intersection of two routing regions, the wiring shape of the net, *i* or *j*, must be detoured from a *top-L*(*bottom-L)* direction to a *detour-top-L(detour-bottom-L)* direction.

For the determination of the wiring directions of all the routing nets in a single layer, a sequential assignment process can be proposed as follows: Based on the classification of the diagonal and adjacent intersections, the wiring directions of all the routing nets on the *D1-type* diagonal intersection relations and some routing nets on the *D2-type* diagonal intersection relations and the adjacent intersection relations can be firstly assigned as *top-L* or *bottom-L* directions. Furthermore, if there is any unassigned routing net on the *D2-type* diagonal, adjacent, crossing and covering intersection relations, the wiring direction of the routing net can be assigned as a *top-L* or *bottom-L* direction. Next, the wiring directions of the assigned routing nets on the *D3-type* diagonal intersection relations can be assigned as feasible *top-L* or *bottom-L* directions under its assignment constraint. Finally, the wiring directions of some routing nets on the crossing intersection relations must be reassigned as *detour-top-L* or *detour-bottom-L* directions.

Refer to the intersection graph in Fig. 3(b), based on the classification of the diagonal and adjacent intersections, the relations on the edges in the intersection graph can be further assigned as illustrated in Fig. 7(a). By using a sequential assignment process, the wiring directions of the 10 routing nets in the intersection graph can be sequentially assigned as "*T*(*top-L*)", "*B*(*bottom-L*)", "*DT*(*detour-top-L*)" or "*DB*(*detour-bottom-L*)" directions on the corresponding vertices as illustrated in Fig. 7(a). For the diagonal and adjacent intersection relations on $e_{1,2}$, $e_{3,4}$, $e_{4,5}$, $e_{4,9}$, $e_{4,10}$, $e_{5,8}$, $e_{5,9}$, $e_{6,7}$, $e_{6,8}$, $e_{6,9}$, $e_{7,8}$, $e_{7,9}$ and $e_{9,10}$, the wiring directions of the nets, 1, 5, 7, 8 and 9, are assigned as *top-L* directions and the wiring directions of the nets, 2, 3, 4, 6 and 10, are assigned as *bottom-L* directions. For the crossing intersection relations on $e_{5,6}$ and $e_{5,7}$, the wiring direction of the net, 5, is reassigned as *top-detour-L* direction. Based on the assignment of the wiring directions of the 10 routing nets, the virtual wiring paths of the 10 routing nets can be illustrated in Fig. 7(b).



(a) (b)

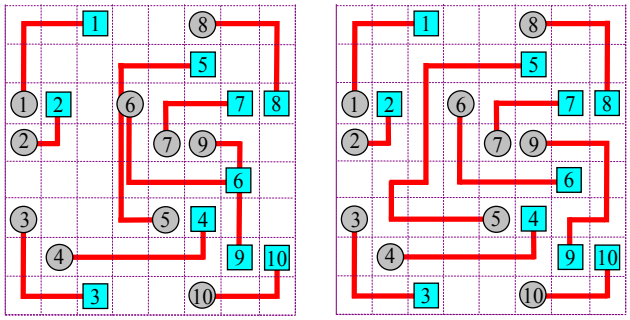
Fig. 7 Assignment of the wiring directions for 10 routing nets

3.2 Region Assignment for Paths of Routing Nets

For an intersection graph, the wiring directions of all the routing nets corresponding to the vertices can be assigned by using a sequential assignment process. Initially, the L-type paths of all the routing nets with *top-L* or *bottom-L* directions and the detoured paths of all the routing nets with *detour-top-L* or *detour-bottom-L* directions can be initially assigned onto routing grids. If some routing grids are used to assign at least two paths, the assigned paths overlapping the same grids must be sorted according to the covering ranges of the paths and further diffused to available grids. If the assigned paths are reassigned onto available grids, the diffusion-based paths will be routed in a single layer.

Refer to the refined intersection graph in Fig. 7(a), based on the assignment of the wiring directions of the 10 routing nets, the L-type paths of the 9 routing nets, 1, 2, 3, 4, 6, 7, 8, 9

and 10, with *top-L* or *bottom-L* directions and the detoured path of the routing net, 5, with a detour-top-L direction can be initially assigned onto routing grids as illustrated in Fig. 8(a). Clearly, the paths of the nets, 5 and 6, and the paths of the nets, 6 and 9, are assigned onto the same grids. Hence, the paths of the nets, 5 and 9, must be diffused onto available grids to maintain single-layer routing as illustrated in Fig. 8(b).



(a) (b)

Fig. 8 Assignment of diffusion-based paths for 10 routing nets

After completing the region assignment for the paths of all the routing nets, the L-type or diffusion-based paths of the routing nets cab be obtains. To minimize the total wirelength of the assigned paths of all the routing nets, the path detours of the routing nets must be sequentially eliminated. Refer to the assigned paths of the routing nets in Fig. 8(b), it is obvious that the path of the routing net, 5, is a detouring path. To minimize the wirelength of the path of the routing net, 5, the path detour of the routing net must be eliminated. As a result, the final path assignment of the 10 routing nets can be obtained and the total wirelength of the 10 given nets can be minimized as 55 as illustrated in Fig. 1(b).

IV. Experimental Results

In single-layer routing, our proposed region-assignment-based approach has been implemented by using standard C++ language and run on a Pentium QuadCore CPU 2.66 GHz machine with 2GB memory. 6 tested examples, *Ex01*, *Ex02*, *Ex03*, *Ex04*, *Ex05* and *Ex06*, are randomly generated in a single-layer gridded routing plane and listed in Table I. In this Table, "*#Routing Grids*" denotes the dimension of the routing grids in a single-layer routing plane, "*#Routing Nets*" denotes the number of the given two-terminal routing nets, "*#Routed Nets*" is the number of the routed nets in a single layer, "*Total Wirelength*" is the total wirelength of the routed nets and "*CPU Time*" is the used CPU time of completing the routed nets in a single layer.

The experimental results show that our proposed approach can route 99.98% of the given nets in single-layer routing for 6 tested examples in reasonable CPU time on the average.

V. Conclusions

Based on the routing constraints on different intersection conditions of two routing regions in a single layer, the wiring directions of the given nets can be assigned. Furthermore, based on the assigned directions of the given nets, the wiring paths of the given nets onto the routing grids can be assigned by diffusing the overlapping paths and eliminating the unnecessary detours in single-layer routing.

References

[1] P. Dehkordi and D. Bouldin, "Design for packageability: the impact of bonding technology on the size and layout of VLSI dies," *Multi-chip Module Conference*, pp.153-159, 1993.

[2] J. W. Fang, C. H. Hsu and Y. W. Chang, "An integer linear programming based routing algorithm for flip-chip design," *Design automation Conference*, pp. 606–611, 2007.

[3] P. W. Lee, C. W. Lin and Y. W. Chang, "An efficient pre-assignment routing algorithm for flip-chip designs," *International Conference on Computer-Aided Design,* pp. 239-244, 2009.

[4] J. T. Yan and Z. W. Chen, "Pre-assignment area-IO RDL routing via extraction of maximal net sequence," *International Conference on Computer Design,* pp.65-70, 2011.

[5] R. S. Wang, R. Shi and C. K.. Cheng, "Layer minimization of escape routing in area array packaging," *International Conference on Computer-Aided Design,* pp. 815-819, 2006.

[6] R. Shi and C. K. Cheng, "Efficient escape routing for hexagonal array of high density I/Os," *Design Automation Conference,* pp.1003-1008, 2006.

[7] C. C. Tsai, C. M. Wng and S. J. Chen, "NEWS: A net-even-wiring system for routing on a multilayer PGA package," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 17, no. 2, pp.182–189, 1998

[8] Y. Kubo and A. Takahashi, "A global routing method for 2-layer ball grid packages," *International Symposium on Physical Design*, pp.36–43, 2005.

[9] S. Liu, G. Chen, T. T. Jing, L. He, R. Dutta and X. L. Hong, "Diffusion-driven congestion reduction for substrate topological routing," *International Symposium on Physical Design*, pp.175–180, 2009.

[10] S. Liu, G. Chen, T. T. Jing, L. He, T. Zhang, R. Dutta and X. L. Hong, "Substrate topological routing for high-density packages," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 2, pp.207–216, 2009.

Table I Experimental results of our proposed region-assignment-based algorithm in single-layer routing

Examples	*#Routing Grids*	*#Routing Nets*	*#Routed Nets*	*Total Wirelength*	*CPU Time(s)*
Ex01	*8x8*	*10*	*10(100.00%)*	*55*	*0.07*
Ex02	*16x16*	*41*	*41(100.00%)*	*229*	*0.21*
Ex03	*32x32*	*169*	*169(100.00%)*	*987*	*0.56*
Ex04	*64x64*	*637*	*636(99.84%)*	*3963*	*2.09*
Ex05	*128x128*	*2418*	*2416(99.92%)*	*15639*	*5.73*
Ex06	*256x256*	*9758*	*9758(100.00%)*	*61984*	*13.78*