Layer Count Reduction for Area Array Escape Routing

Rui Shi, Hongyu Chen, Chung-Kuan Cheng
Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive
La Jolla, CA, 92093, USA
rshi@cs.ucsd.edu, hchen@cs.ucsd.edu,
kuan@cs.ucsd.edu

Dan Beckman, Dawei Huang
SUN Microsystems
La Jolla, CA, USA
Dan.Beckman@SUN.COM, Dawei.Huang@sun.com

Abstract

Escape routing for area array packaging becomes more complicated as the I/O pin count increases. An efficient way to break out the I/O pins in the array can reduce the number of escape routing layers and result in a low manufacturing cost. We analyze the influence of pins escape sequence to the layer count and analyze the escape bottleneck in an area array using maximum flow algorithm. We implement an automatic escape routing program and compare four escape sequence strategies. Finally we identify that the two-sided method is the most effective way to reduce the number of escape routing layers.

Keywords: escape routing, area array, escape sequence, layer count reduction

I. Introduction

With the steady progress of high performance electronic systems, the number of chip/package input/output (I/O) pins has continuously been growing. Increasing demand for high I/O pins count prompts the packaging industry to explore new technologies, such as area array interconnection [1], CSP (chip scale packing), BGA (ball grid array) [2], and so on. Area array packaging contains an array of pads, pins, or solder balls located directly underneath a unit of package area. In order to connect the area array I/Os to the next level assembly, all the I/O pins need to break out to the outside. As shown in Fig.1, the wires for breaking out I/O pins are referred to as Escape Routing [3][4]. Currently, the high count and density of I/Os require multilayer for escape routing. These interconnections directly affect the cost and performance of the electronic systems.

The higher I/O counts will require an increase in the number of escape routing layers and result in a higher manufacturing cost of substrates used as a package or a board. Reduction of the layer count will become an efficient way to achieve high performance packaging with low cost. Intuitively some dimensional changes will reduce the number of escape routing layers, such as decreasing the width of wires, and the space between wires. However in these cases, the cost, yield and reliability will become issues and new process technology will be necessary as well. An alternative method is to design the I/O locations. Gasparini et al. [5] have shown the specific placement of bumps can contribute to the routability while it achieves the reduction of layers count with change in footprint.

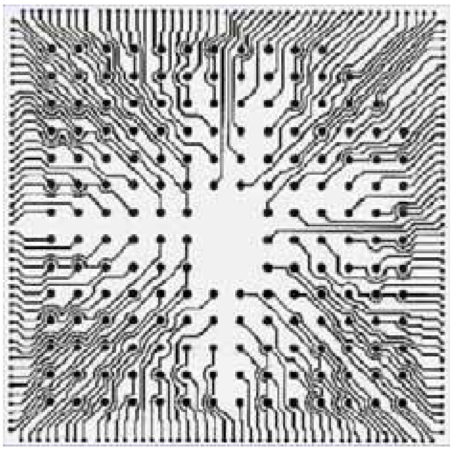


Figure 1 Escape routing connects I/O pins to outside

Methods to reduce the layer count without making any change in manufacturing process and footprint standard are accordingly desired. The conventional escape routing methods for an area array will break out pins in order from outside rows to inner rows. Horiuchi et al. [7] suggest a preferential escape routing method, which creates triangular pad geometry resulting in a higher wiring efficiency compared with the traditional method. This method explores the diagonal spaces for routing while its efficiency will decrease along with the shrinking of the array size.

In this paper, we analyze the relation between escape routing layer count and escape sequence. We implement the maximum flow algorithm to analyze the escape bottleneck in area array. Based on our analysis, we implement an automatic escape routing program to compare different strategies for pins escaped sequence and identify that the two-sided method is the most effective way to reduce the number of escape routing layers.

The rest of the paper is organized as follows. Section II describes the escape routing problem and defines the technical parameters. The relation between layer count and escape sequence is analyzed in section III. Section IV explains the automatic escape routing algorithm and escape sequence strategies. Section V lists the experimental results. And we discuss our conclusions from this research in section VI.

II. Problem Description

The I/O pins area array we consider is a regular array, which is a fully populated array and is a typical square grid matrix. We assume the blind via technology has been used, which means that the escaped pins/pads/vias will disappear in the following routing layers.

We define size n to represent the dimension of the regular I/O pins area array. Thus there are n^2 pins in a populated regular area array of size n. For escape routing, these pins are the objects; the corresponding pads are the obstacles and the spaces scattering among the pads are the resources. The related technical parameters describing the I/O pins area array include the pitch (P) between two adjacent pads, the diameter (D) of one pad, the width (W) of escape wires, the minimum space (S_p) between the escape wire and the pad and the minimum space (S_w) between two adjacent escape wires, as shown in Fig.2.

One of the main objectives of escape routing is to decrease the number of routing layers used for breaking out all I/O pins in the area array, which will affect the manufacturing cost directly.

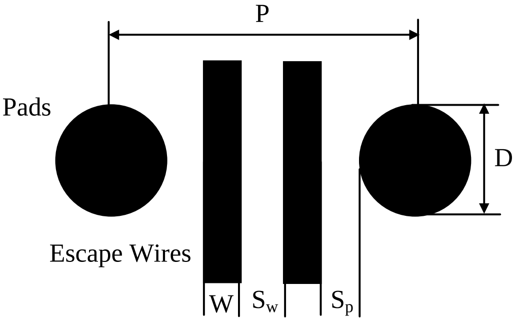


Figure 2 Technical parameters describe area array

III. Escape Sequence Analysis

2.1 Escape Channel

The space scattering among the pads will determine the number of escape wires which can go through in a single layer. Fig.3 shows three different kinds of escape channel. Channel I represents the routing space between two adjacent pads in the same row or column of the array. Channel II represents the routing space between two adjacent diagonal pads. Channel III represents the routing space between a pad and a routing wire. The number of escape wires which can go through these channels is defined respectively as

$$\alpha_I = \frac{P - D - 2S_P + S_w}{W + S_w} \tag{1}$$

$$\alpha_{IT} = \frac{\sqrt{2}P - D - 2S_P + S_w}{W + S_w}, \tag{2}$$

$$\alpha_{III} = \frac{P - 0.5D - 0.5W - S_P}{W + S_w}. \tag{3}$$

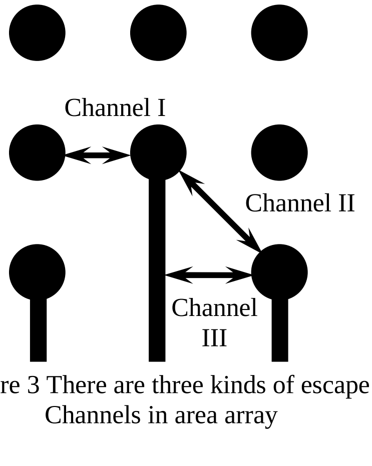


Figure 3 There are three kinds of escape Channels in area array

According to the data about assembly and packaging provided by ITRS (International

Technology Roadmap for Semiconductors) [8], the pad size is usually 30%-60% of pad pitch and the minimum space (S_p) between wire and pad is usually equal to the minimum inter wire space (S_w). Table 1 shows the number of escape wires for different channels using two groups of typical data.

Table 1. Compare the number of wires going through three kinds of channels

	BGA/FBGA/CSP	**Flip Chip**
Pad Pitch (μm)	400	150
Pad Size (μm)	160	75
Line Width (μm)	48	20.4
Line Spacing (μm)	48	20.5
α_I	2	1.33
α_{II}	3.73	2.85
α_{III}	2.58	2

Obviously, channel II and III can allow more wires to go through and therefore more these kinds of channels in the array will be beneficial to escape more wires in one routing layer. Thus in order to reduce the number of layers used for escape routing, we can choose some escape sequence to form those kinds of channels.

2.2 Escape Bottleneck

Actually the escape routing for an I/O pins area array is some kind of area routing. We construct a routing graph G = (V, E) to extract the escape routing resources in an array. The maximum flow solution for the routing graph will reveal the bottleneck location of escape routing.

2.2.1 Maximum Flow Problem Formulation

We map the I/O pins area array into grid with pins on the crossing points of the grid, as shown in Fig.4. Each grid cell corresponds to the routing space surrounded by four I/O pins on its corners and is represented by a vertex. The edge connecting the vertices represents the routing channel between the adjacent grid cells. For a regular I/O pins area array, the routing graph looks like a mesh.

The vertices and edges are attached with a capacity, which is the number of available routing wires. The capacity of an edge is defined as the maximum number of wires that can route through two adjacent pads, corresponding to channel I. Similarly, the capacity of a vertex is defined as the maximum number of wires that can route through the grid cell, corresponding to channel II. If some pins do not exist, which means some grid cell doesn't have exactly four pins in its corners, and then we will assume one routing wire locates at that corner, similar to channel III.

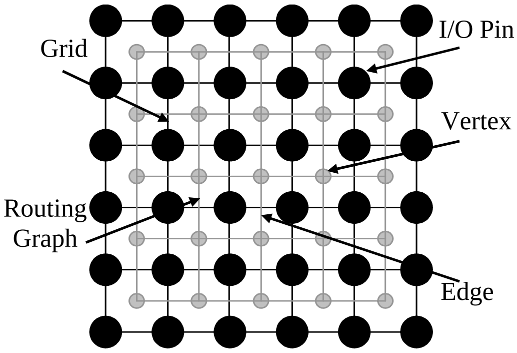


Figure 4 The routing graph represents the routing resource in a regular area array

We add sink vertices, which are located on the boundary of the array, to represent the destinations of escape routing and all sinks will be connected together by a hyper-sink-vertex (Fig.5). The I/O pins in the array are added to the routing graph as source vertices and similarly there's a hyper-source-vertex (Fig.5).

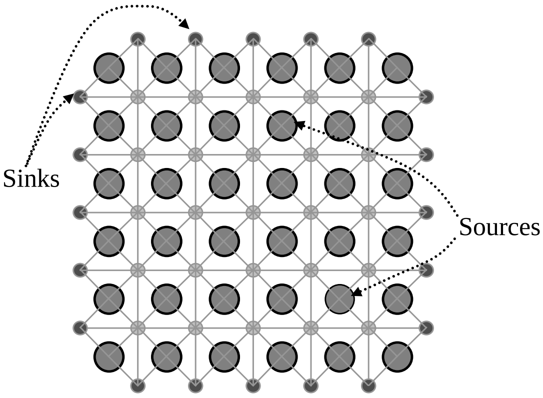


Figure 5 Add sink and source vertices to the routing graph

2.2.2 Escape Bottleneck Analysis

We implement the maximum flow algorithm for the routing graph we formulate. According to the solutions, we can identify the escape bottleneck in the area array and can get hints about escape sequence strategies.

We define bottleneck edge as the edge in the routing graph whose flow is equal to its capacity in the maximum flow solution. The bottleneck contour formed by all the bottleneck edges will constrain the maximum number of routing wires escaping in one layer.

We do experiments on different distribution patterns of the I/O pins area array and identify all the bottleneck edges. Fig.6 shows some results. It's a 20x20 array and the pad pitch, pad diameter, line width, and line spacing are 150µm, 75µm, 20µm, and 20µm respectively. We use red edges to represent bottleneck edges.

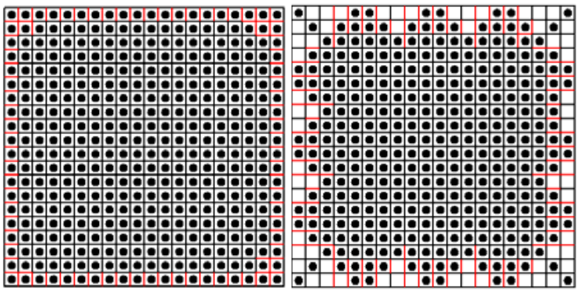


Figure 6 The escape bottleneck exists on the array boundary

The experimental results reveal that the bottleneck contour of the area array follows the outline of the array. Thus the number of I/O pins that can be break out in one routing layer will be constrained by the capacity of the array outline. Obviously the capacity of the indented array outline, which contains more channels belonging to channel II and III, is larger than that of the regular square outline, as shown in Fig.6. Thus in order to reduce the number of layers used for escape routing, we can choose some escape sequence to form the indented array outline.

IV. Escape Sequence Strategies

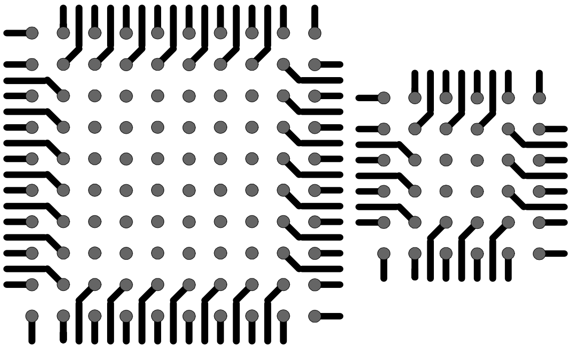
Intuitively we try to break out I/O pins for one routing layer as many as possible to decrease the total number of layers. According to the analysis in section III, the escape sequence, which can form more channels of channel II and III and can form indented outline, will potentially increase the number of I/O pins escaped in one routing layer. So the I/O pins should be escaped following some indented way. Also the I/O pins escaped in previous layers will shape the array outline for the next layers, so the I/O pins in the outside rows should not be escaped very early, otherwise the array outline will shrink very fast.

4.1 Escape Sequence

According to the analysis, we consider the following four different pin escape sequence strategies.

4.1.1 Row-by-row Sequence

It is the conventional approach, as shown in Fig.7, which breaks out pins row by row from outside to inside. Suppose we can escape one wire between two adjacent pads, then we can escape two rows of I/O pins on each layer and we need n/4 escape routing layers for a regular I/O pins area array with size n. In this method, the array outline shrinks layer by layer quickly.

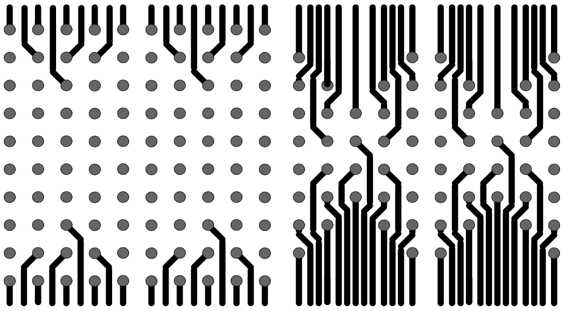


(a) First layer (b) Second layer

Figure 7 Row-by-row escaped sequence

4.1.2 Parallel Triangular Sequence [7]

This method divides the pins into groups and breaks out each group with a triangular outline, as shown in Fig.8. It allows more pins breakout because it have more channels of channel II and III. Compared with the traditional escape sequence, in this sequence the capacity of one routing layer is increased because the indented outline. But the array outline will also shrink layer by layer quickly.

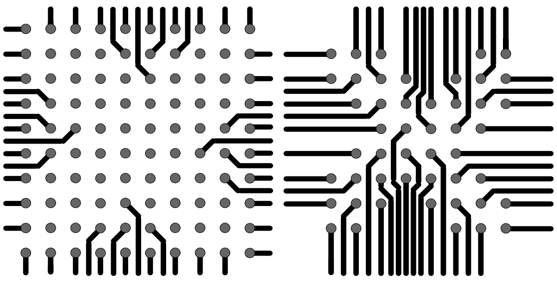


(a) First layer (b) Second layer

Figure 8 Parallel triangular escape sequence

4.1.3 Central Triangular Sequence

It breaks out pins from the center of the outside row and expands the indent with a single triangular outline, as shown in Fig.9. In this method, the capacity of escape routing is increased continuously layer by layer while the capacity for the first several layers is small.

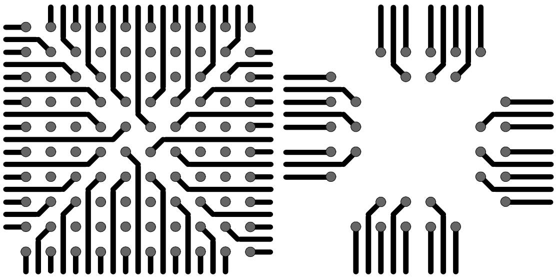


(a) First layer (b) Second layer

Figure 9 Central triangular escaped sequence

4.1.4 Two-sided Sequence

This approach breaks out pins from the inside as well as from the outside, as shown in Fig.10. The outline shrinks slowly and also follow indented shape. For a proper arrangement, we can have a very even number of pins to break out on each routing layer. This can efficiently decrease the number of layers for escape routing.



(a) First layer (b) Second layer

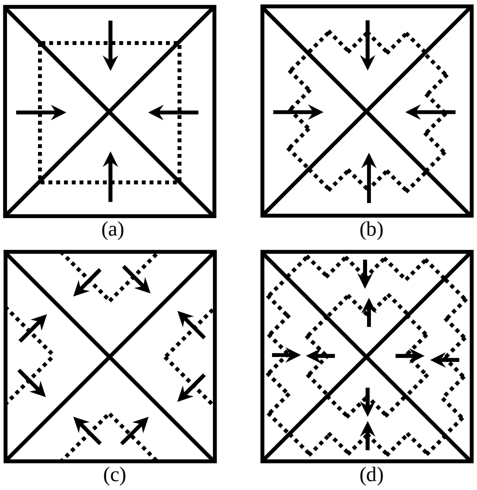
Figure 10 Two-sided escaped sequence

Fig.11 shows the global view of the movement of breakout outline for these four strategies.

We implement automatic escape routing program using a river routing approach [6].

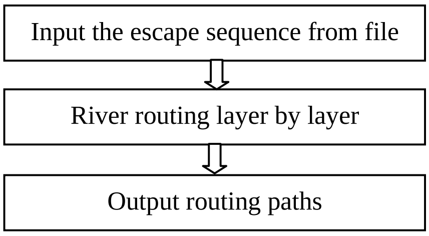
4.2 Escape Routing Algorithm

Fig.12 shows the flow for our escape routing algorithm. The escape sequence will be inputted from description file. We execute the escape routing for each pin using a river routing approach. In order to utilize diagonal wires, we use octagon to approximate the figures of the pads. Thus the escape routing wires will be composed of segments with 0°, 90°, 45° and 135°. For each pin breakout routing, we follow the profile of previous routing wire and consider the surrounding octagonal obstacles. Finally, the routing paths, the number of routing layers and the number of I/O pins escaped on each layer will be reported to output files if the input sequence is feasible.



(a) (b) (c) (d)

Figure 11 The global view of breakout outline



Input the escape sequence from file

River routing layer by layer

Output routing paths

Figure 12 Escape routing algorithm flow

V. Experimental Results

We break out a regular I/O pins area array with size n=40 and n=20. And the pad pitch, pad diameter, line width, and line spacing are 150μm, 75μm, 20μm, and 20μm respectively. Table 2 and 3 lists the number of I/O pins escaped on each layer for those four escaped sequence strategies for 20x20 and 40x40 array respectively. Comparing the results of 40x40 area array, we can see the characteristic for those four escape sequence strategies. For the traditional row-by-row method, the number of breakout pins decreases rapidly layer by layer. It requires 10 total layers. For the parallel triangular method, the number of breakout pins on one-layer peaks at the second layer and drops slowly afterwards. This method reduces the required number of layers from 10 to 7. For the central triangular method, the number of breakout pins starts from small value but keeps on increasing and thus reduces the number of layers to 6. The two-sided method is the most efficient, which achieves the result in only 5 routing layers.

Table 2 Compare the number of breakout pins escaped on each layer for four sequences 20x20 area array

Layer	**Row by row**	**Parallel triangular**	**Central triangular**	**Two sided**
1	144	132	92	140
2	112	144	116	160
3	80	96	140	100
4	48	28	52	-
5	16	-	-	-

Table 3 Compare the number of breakout pins escaped on each layer for four sequences 40x40 area array

Layer	**Row by row**	**Parallel triangular**	**Central triangular**	**Two sided**
1	304	276	100	312
2	272	340	156	328
3	240	292	228	308
4	208	240	300	324
5	176	164	372	328
6	144	124	444	-
7	112	164	-	-
8	80	-	-	-
9	48	-	-	-
10	16	-	-	-

VI. Conclusions and Future Works

In this paper, we point out the importance of the pin escape sequence for area array escape routing. The layer count can be decreased by optimizing the pin escape sequence. We implement maximum flow program to analyze the escape bottleneck in area array and implement automatic escape routing program to compare different pin escape sequence strategies. The two-sided escape method provides an efficient way on reducing the layer count.

In our algorithm, we concentrate on the reduction of layer counts. The signal integrity will be introduced to the objective in the next step and we also can explore special operations for special signal pins, such as power/ground, differential signals, and so on.

Acknowledgement

The authors would like to thank the anonymous reviews for their constructive suggestions. Rui Shi, Hongyu Chen and C.K. Cheng also would like to thank the support from NSF grant MIP-9987678 and California MICRO program.

References

Presented Paper

[1] Chung TC, Ghahghahi F, Oberlin B, Carey D, Nelson D, “Area Array Packaging Technologies for High-Performance Computer Workstations and Multiprocessors,” The proceedings of 46th Electronic Components and Technology Conference, 1996, pp.902-10.

[2] Man-Fai Yu, Wei-Ming Dai, “Single-Layer Fanout Routing and Routability Analysis for Ball Grid Arrays,” IEEE/ACM International Conference on Computer-Aided Design. Digest of Technical Papers, 1995, pp.581-6.

[3] Winkler E, “Escape Routing from Chip Scale Packages,” 9th IEEE/CPMT International Electronics Manufacturing Technology Symposium, 1996, pp.393-401.

[4] Guinn KV, Frye RC, “Flip-Chip and Chip-Scale I/O Density Requirements and Printed Wiring Board Capabilities,” 47th Electronic Components and Technology Conference Proceedings, 1997, pp.649-55.

[5] Gasparini NM, Bhattacharyya BK, “A Method of Designing a Group of Bumps for C4 Packages to Maximize the Number of Bumps and Minimize the Number of Package Layers,” 44th Electronic Components and Technology Conference Proceedings, 1994, pp.695-9.

[6] Hsu C-P, “General River Routing Algorithm,” ACM IEEE 20th Design Automation Conference Proceedings, 1983, pp.578-83.

Journal Article

[7] Horiuchi M, Yoda E, Takeuchi Y, “Escape Routing Design to Reduce the Number of Layers in Area Array Packaging,” IEEE Transactions on Advanced Packaging, vol.23, no.4, Nov. 2000, pp.686-91.

Book

[8] International technology roadmap for semiconductors, 2002 edition.