Layer Minimization of Escape Routing in Area Array Packaging

Renshen Wang   Rui Shi   Chung-Kuan Cheng

Department of Computer Science and Engineering, University of California, San Diego

La Jolla, CA 92093-0404

rewang@cs.ucsd.edu   rshi@cs.ucsd.edu   ckcheng@ucsd.edu

ABSTRACT

We devise a central triangular sequence to minimize the escape routing layers in area array packaging. We use a network flow model to analyze the bottleneck of the routable pins. The triangular patterns are generated in a reverse order from the last to the first layer. We demonstrate that the triangular pin sequence maximizes the sum of escape pins in the accumulated layers and thus minimize the number of escape routing layers. A test case is presented to illustrate the approach.

Keywords

Escape routing, bottleneck analysis, central triangular pattern.

1. INTRODUCTION

Packaging is one important process to enhance chip communication performance and reduce production cost. According to Rent's rule [1], the number of I/O signals of a module is a function of the number of gates in it:

$$N_p = K_p N_g^{\beta}$$

where N_p is the number of external signal connections, N_g is the number of logic gates, K_p is a constant and β is the Rent's rule constant which depends on circuit. To accommodate the growing number of I/O signals N_p, area array interconnection is widely used between die to packaging, chip scale packaging (CSP) to board, or ball grid array (BGA) to board interconnections.

Escape routing connects the I/O array to the next level assembly [2]. Single layer escape routing in a grid array is elaborately explored in [3]. In [4], pin locations are even adjusted to accommodate the escape wires. However, as N_p increases, multilayer high density substrates are needed. Since the number of escape routing layers affects the manufacture cost, our goal is to reduce the routing layers.

The escape sequences evolve from traditional row by row, parallel triangular indents, to various escape patterns. In [5], Horiuchi et al demonstrated that parallel triangular indents increase the surface of the boundary and thus allow more pins to break away. In [7], Shi et al explored different sequences to further reduce the number of layers.

In this paper, we introduce a strategy for escape routing design to minimize number of layers under some classes of parameters. We first set up a network flow model corresponding to the single layer escape routing problem. We analyze the bottleneck of routable pins on each layer. A central triangular pattern is devised
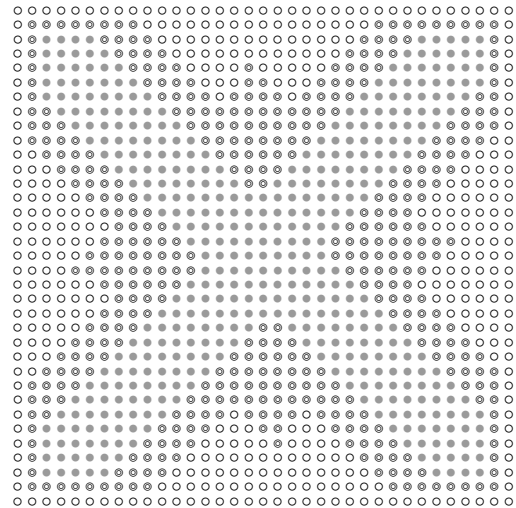
Figure 1. An industrial BGA case

to distribute the pins to proper layers. We demonstrate that in a reverse order of routing layers from last to first, the sum of escaped pins is maximum in accumulated layers for a class of parameters. Moreover, this class of parameters covers most of industrial cases. In figure 1, a 35×35 BGA pin array is shown to be broken out in 3 layers, where the pins in the first layer are "○", second layer "◎", and third layer "●". The 3-layer solution is the lower bound for this case, while by the traditional ring-by-ring approach [2] we need 6 layers instead.

The rest of this paper is organized as follows. In section 2, we describe the problem formulation. Section 3 analyzes the bottleneck of routable pins by using a network flow model and then proves the upper bound of the number of routable pins in each layer. Then section 4 introduces our "central triangular pattern" for the multilayer escape routing solution. Finally in section 5, we illustrate the method with a detailed example. Section 6 concludes the paper.

2. PROBLEM FORMULATION

In this paper, we consider array of pins in regular square grids. Given an $n \times n$ pad matrix, we break out all pads to the outside region. We denote pad pitch p, pad diameter d, line width w, and line spacing s. The objective is to minimize the number of escape routing layers.

We arrange the sequence of the break-out pins inside the boundary. The pins on the boundary are always broken out on the current layers. For the pins inside the boundary, we use routing channels between the grids to break out. There are two types of channels: grid channel and diagonal channel [Fig. 2]. A grid channel is the routing space between two adjacent pads on the same row or column. The capacity of the channel is equal to the maximum number of available wires in the dimension [Fig. 2]:

$$C = \lfloor (p - d - s)/(w + s) \rfloor$$

where $\lfloor x \rfloor$ denotes the floor, i.e. the maximum integer no larger than x. A diagonal channel is the routing space between two pins adjacent at corners. The capacity of the channel is equal to

$$D = \lfloor (\sqrt{2}p - d - s)/(w + s) \rfloor$$

For instance, with a set of parameters 225/100/25/25 (for $p/d/w/s$), we have channel capacities, $C=2$ and $D=3$. With parameters 350/200/50/50, we have $C=1$ and $D=2$.
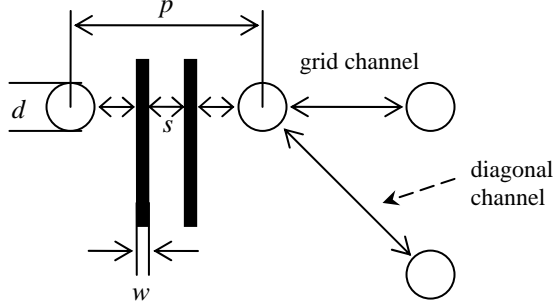

Figure 2. Parameters and channels

We assume that the blind via technology is used. In other words, a break-out pad will disappear in the following routing layers. Thus, the space left by previous escaped pins can be used to route more wires. We also assume that all wires are composed of segments horizontal, vertical or at 45°, 135° angles.

3. SINGLE LAYER NETWORK FLOW MODEL AND BOTTLENECK ANALYSIS

3.1 Pad Matrix to Network Flow

We follow the network flow model in [6] to represent the routing problem. We construct a network graph as follows: Add an edge from source S to each pin which needs to be broken out, and add an edge from each channel end outside the matrix to termination T. Between S and T, we have nodes and edges representing the joints and channels [Fig. 3]. Between the four adjacent pads, we use edges with capacity C to represent the grid channels. We use 3×3 meshes with edge capacity $\lceil D/3 \rceil$ to represent diagonal channels. The mesh is connected to the grid channel at four corners [Fig. 3].

After some pins break out, we update the network by revising the capacity of the edges. For example, suppose pad IV in figure 3 escaped. The pad space can be used by more wires. Thus, we add capacity $\lceil d/2(w+s) \rceil$ to the edges beside pad IV as an upper bound. Note that possibly fractions of original grid channels are combined together into the new big channel and produce additional capacity.

In summary, we analyze the single layer escape routing problem by constructing a graph and using a network max-flow algorithm.
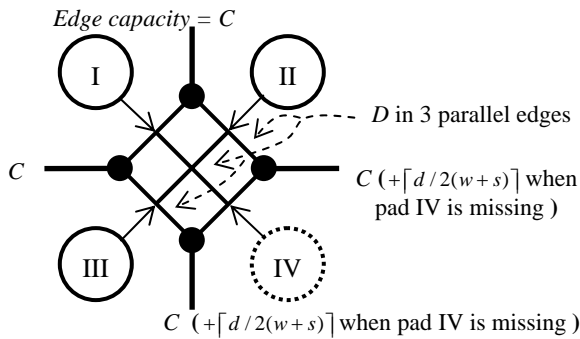

Figure 3. A graph component between 4 pads

The max flow provides an upper bound of the routing capacity and global guide for the escape routing.
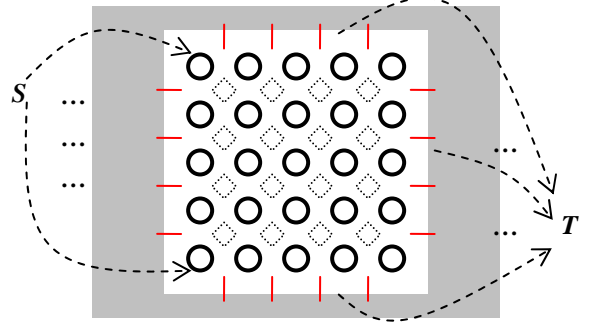

Figure 4. A small pad matrix example

3.2 Bottleneck of Breaking out Pins

Figure 4 depicts a 5×5 array of pads and its escape routing graph, where each graph component is represented by a "◇". When the matrix is full, the bottleneck of the escape routing is the channels at the boundary of the array. Therefore, the number of break-out pins in the first layer is restricted by the total capacity of $4(n-1)$ channels on the four sides of the matrix. This bottleneck can be derived from the network model.

The popular ring-by-ring method is inefficient because the boundary of the array shrinks quickly layer by layer. The grey part of figure 4 can be the space left by previous pin pads. The total region reduces and the dimension of the four sides also drops. Since fewer pins can be broken away in later layers, more layers are needed to finish the pin breakaway.

The concept of "bottleneck" can help us to list some necessary conditions for "full space utilization":

- On the 1st layer, the channels at the boundary are full.
- On the kth layer (k>1), the kth outline is full and the empty space left by the pins broken in previous layers remains to be bottleneck area.

The second conditions may be impossible to satisfy everywhere in all cases. But generally, if most part of usable space is not wasted, our solution will be relatively good. In figure 5(a), the previous break-out pins distribute wide at the bottom of the matrix. The breakaway pins leave an open space wide at the bottom. However, inside the array the channels connected to the opening are still limited by the existing pins. Consequently, the wide opening at the bottom does not help much. In figure 5(b), the pins previously escaped distribute as a narrow rectangular opening. The bottleneck remains at the bottom of the array. The space at the top corners of the rectangular opening does not help. The opening at these corners is wasted. Finally we look at figure 5(c), which is a possible solution. The escaped pins form an isosceles triangle, where the channel grows wider from inside to outside. For this pattern, extra wires stream into the triangle as we go down the
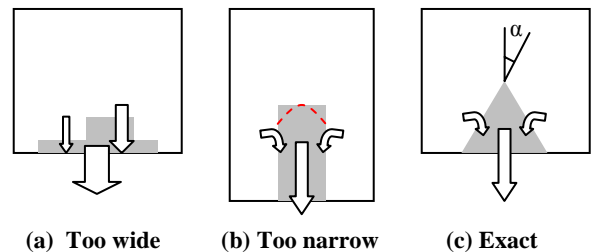

| (a) Too wide | (b) Too narrow | (c) Exact |

Figure 5. Different patterns of channel space

rows as the capacity grows linearly to the final width. Each part of the opening is fully filled with wires. Therefore the space efficiency is the highest among the three patterns in figure 5.

From figure 5, we observe that a triangular shape is relatively efficient for breaking out pins. The top angle of this triangle should be 2α where α is approximated as

$$\alpha \approx \arctan \frac{C+1}{p/(w+s)+C-D}$$

because in this way, each time the opening increases by 1 pin at each side, it moves outward by cotα pins, passing one diagonal channel and (cotα–1) grid channels, with (cotα + D + C(cotα–1)) ≈ $p/(w+s)$ more lines going out. This is the exact increased capacity of 1 pin, so with this angle α, the triangular channel space can achieve full efficiency.

In industrial cases, p is usually close to $(D+1)(w+s)$, resulting in tanα ≈ 1. In the sequel, we assume that the angle α=45° to simplify the description. Hence the top angles of our triangles in the "triangular pattern" are all 90°. For other values of α, the general properties could be similar. However, further research is needed to optimize the results.

We also notice that triangle is not the only shape which can use space efficiently. The necessary component is the slanting boundaries with angle 2α between adjacent segments which will be discussed in the following section.
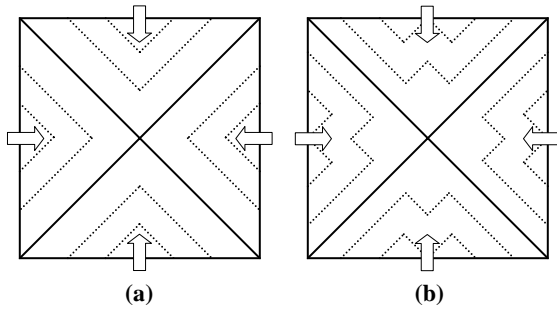

Figure 6. Different triangular patterns

4. CENTRAL TRIANGULAR PATTERN

Our goal is to assign pin breakaway sequence so that the number of routable pins is maximized in a given number of layers. When the number of breakaway pins reaches n^2, the sequence becomes an optimal solution.

The strategy is that the breakaway pins should leave triangular opening for the following layers. Since we assume that the angle of the optimal triangles is 90°, we can divide the array into four triangular regions as shown in figure 6. We only have to focus on one region because the four regions are identical in shape and size.

Though the shape of the triangles we use is decided now, the size and number of these triangles remains unknown. We can make single triangles of escaped pins and enlarge the triangle layer by layer as shown in figure 6(a). An alternative is to distribute the pattern into a few smaller triangles as shown in figure 6(b). In this way the total channel capacity is larger in the first several layers.

4.1 Maximum Routable Pins in the Last Layer

We first find out the upper bound of routable pins in the last layer. We then try to achieve the maximum in the accumulated layers in an order from last to first. In the following derivation, we assume $p = (D+1)(w+s)$ so that the empty space is fully utilized in 90° triangles. We can assume $D \leq 2C$, otherwise the diagonal capacity is limited by the two adjacent grid channels, $2C$. Let us denote the last layer be the k'th and $m = n - 2k + 2$.

Claim 1: With parameters of $p = (D+1)(w+s)$ and $D \leq 2C$, the number of routable pins in the last layer is no greater than $4[(D+1)(m-1) - D^2/2 - 3D + 2C]$.
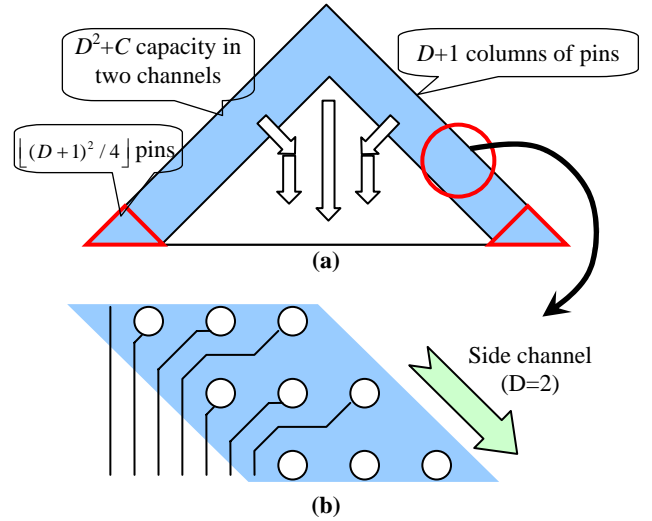

Figure 7. Max number of routable pins

Proof: The maximum width of the opening at the bottom is $(m-1)-2(D+1)$ pads. Because to provide enough lines to achieve this density, we need $(D+1)$ pin pads per row at each side of the triangle like figure 7(a). Further increasing this opening will only reduce the routable pins since there will be not enough pins left to provide the lines filling the space. So there are at most $(D+1)(m-3-2D)+C+2$ lines in the middle open space of the large triangle.

For the two "side channels" shown in figure 7(b), the capacity can be calculated by counting the diagonal channels in them. We first route all the lines like in figure 7(b) so that all the routed wires in the right side channel are pointing to west or south-west. When we need the capacity for the flows towards south-east, this capacity is not affect by existing wires. By combining two "side channels" of adjacent triangles (see the third layer in figure 1), we can count D diagonal channels and 1 grid channel, so the total capacity of the two side channels is D^2+C.

The last part of pins possible to be broken out is at the corners of the pad matrix. By routing the lines as in figure 7(b), there are two small triangles of pins left in the corners and they can be broken out directly. The number of pins in a small triangle is not greater than $(D+1)^2/4$. Therefore, the total upper bound is 4 times the sum on each region, $4[(D+1)(m-1) - D^2/2 - 3D + 2C]$. □

Note that the details in the top corners may vary in order to fully utilize the capacity, but the difference will be small. For the case that p is not exactly $(D+1)(w+s)$, this result only serves as a general guideline for the breakaway.

4.2 Escape Routing by Central Triangular Pattern

We first analyze the upper bound of the number of breakaway pins in the last two layers. We then extend the derivation to the last j layers. The extension ends when $j = k$.

From the k'th layer toward the $(k-i)$th layer, the upper bound of the pin breakaway capacity is reduced in each layer. In figure 8(b), there are two small triangles of pins left in the first layer, which are formed by a big triangle minus the "extra pins area" in figure 8(a). The pins in the "extra area" escape through the "side channels" where the pins are to escape in later layers. The shape of this area

can vary and form different number of small triangles, which can be tuned to break out all the pins as early as possible. From the last layer upward to the first layer, this forms a central triangular pattern which is a combination of the two strategies in figure 6.
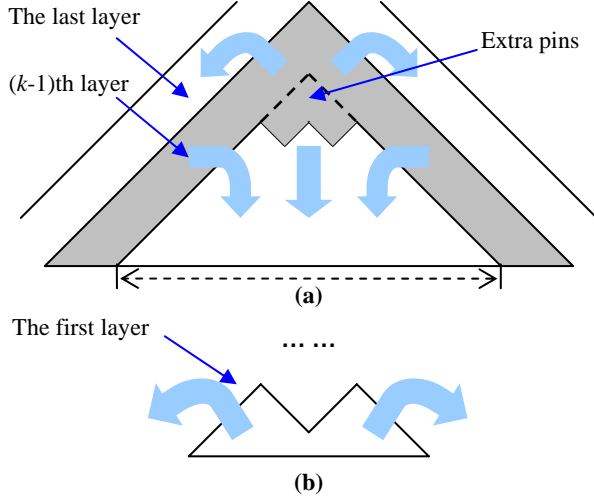

Figure 8. Central triangular pattern method

Claim 2: With the same condition as in claim 1, the total number of routable pins in the last 2 layers is no greater than $4[(D+1)[(D+1)(2m–2–2D) – D^2 – 6D + 4C + D(D+1) + 2(D+1)]$.

Proof: By claim 1, we have the last layer with maximum number of escaped pins. As figure 8(a) shows, the (k-1)th layer has a triangular shape similar to the k'th layer. The open space in the triangle can achieve line density $(D+1)$ per pad, but its width is $2D$ pads less (since the triangle shrinks by $2(D+1)$ but the matrix size increases by 2) than the kth layer, so the number of pins escaping by this triangle is bounded by $[(D+1)(m–1–2D) – D^2/2 – 3D + 2C]$, which can be inferred from claim 1.

Besides, the "extra area" at the top of the triangle can escape through the space of the k'th layer break-out pins. We can count $(D+1)$ diagonal channels on the two sides, with $D(D+1)$ extra capacity. And we have $2(D+1)$ more pins at the bottom of two sides. Summing up the two layers produces the result in the claim.

If we want to increase the number of routable pins in the (k-1)th layer, we have to effectively enlarge the open space. Say we increase the width by one pad, by doing this we delete at least 1 pin. Then we need more pins to fill the increased space. But all the bottlenecks above it are fully filled, the only source of available pins is in the k'th layer. Clearly, bringing pins from the k'th layer to the (k-1)th cannot increase the total number in these two layers. If we want to add more pins into the k'th layer, we have to place them in the mid open space: These pins occupy at least one column of the open space and completely counteract the enlargement we made. Therefore, the enlargement of open space we made in the (k-1)th layer does not bring any benefit for the total number of routable pins in the last two layers. In conclusion, this pattern can break out the maximum number of pins in the last 2 layers.

In the same way, we count the upper bounds of routable pins in the (k-2)th, (k-3)th and earlier layers, which lead to

Claim 3: The upper bound of the total number of pins in the last j layers is the sum

$4\sum_{1\le i\le j}[(D+1)[m–1–2(i–1)D]–D^2/2–3D+2C+(i–1)(D+2)(D+1)]$
$= 4 j[(D+1)[m+j–2 – (j–1)D/2] – D^2/2 – 3D + 2C].$

With this lower bound, we substitute k for j, and the upper bound in the last k layers need to be no less than n^2 in order to break out all the pins in the k-layer solution. Hence we have

Claim 4: The lower bound of k is the smallest integer among the solutions of inequality:

$–2(D+1)(D+2)k^2 + [4(D+1)n–10D+8C]k – n^2 \ge 0$

By the strategy of figure 8, the break-out pins in each layer can almost reach the upper bound, provided that all available channels are fully utilized. Together with the conditions of claim 1, the central triangular pattern can produce a layout with the number of layers approaching the minimum solution.

5. CASE STUDIES

In an industrial FCPBGA case, we have a 35×35 pin array with parameters (225μm, 100μm, 25μm, 25μm). Figure 1 shows the pin distributions on the three layers in our solution. The details of each line are explicitly shown in figure 9. Since all the strategy is centrosymmetric in the four triangles, we draw the solution in one matrix: The first triangle describes the pins in each layer, and the routing details of each layer are in the other three. In this solution, lines on the three layers use up every bottleneck in the matrix, so the number of layers is smaller than all previous methods. Though p=225 does not exactly equal to $(D+1)(w+s)$, simple upper bound calculation shows that 2-layer solution does not exist in this case. In most of other practical cases, the parameters also usually deviate from the exact condition of claim 1 by a small fraction. If we cannot assert in these cases the solution is optimal, we can still guarantee that the solution is very close to the optimal one, e.g. at most one layer more than the minimum.

Besides this case, there are many other parameter sets and array sizes. We list some parameters in the following table, and the corresponding grid channel capacity C and diagonal channel capacity D. The array sizes in current cases are small enough so that the "extra area" [Fig. 8] can always break out.

Table 1. Case parameters (size unit μm)

Case \ Parameters	p	d	w	s	C	D
FCPBGA	225	100	25	25	2	3
FBGA/CSP (2018)	100	40	12	12	2	3
Flip chip-Package	350	200	50	50	1	2
CSP-Board	500	260	80	80	1	2

The FBGA/CSP parameters in 2018 will have smaller dimensions, but surprisingly the channel capacities (C and D) for escape routing remain exactly the same, so our method still applies.

The parameters of flip chip to package and CSP to board in the table are used in [5]. For the 40×40 case, the best result in [5 (Fig. 11)] uses 6 layers. Our method produces a 5 layer solution which is shown in figure 10. Here we only show the distributions, where the 1st, 3rd and 5th layers are marked with black dots and the 2nd, 4th are marked with white circles. The routing details have a similar pattern as figure 9 and so omitted. In fact, 5 layers are enough to break out a 42×42 area array under these parameters.

6. CONCLUSIONS AND FUTURE WORKS

A minimum bound of escape routing layers in grid array is now solved when the parameters satisfy a set of constraints. We show that the central triangular pattern can produce solutions achieving the lower bound when 90° top angle is most efficient. Most constraints are consistent with industrial cases. When the parameters are different, the top angle of the triangles may vary but the basic pattern still applies. The optimal solution remains open when 2α does not exactly divide 360°. However, we can use the proposed strategy as a general guideline.
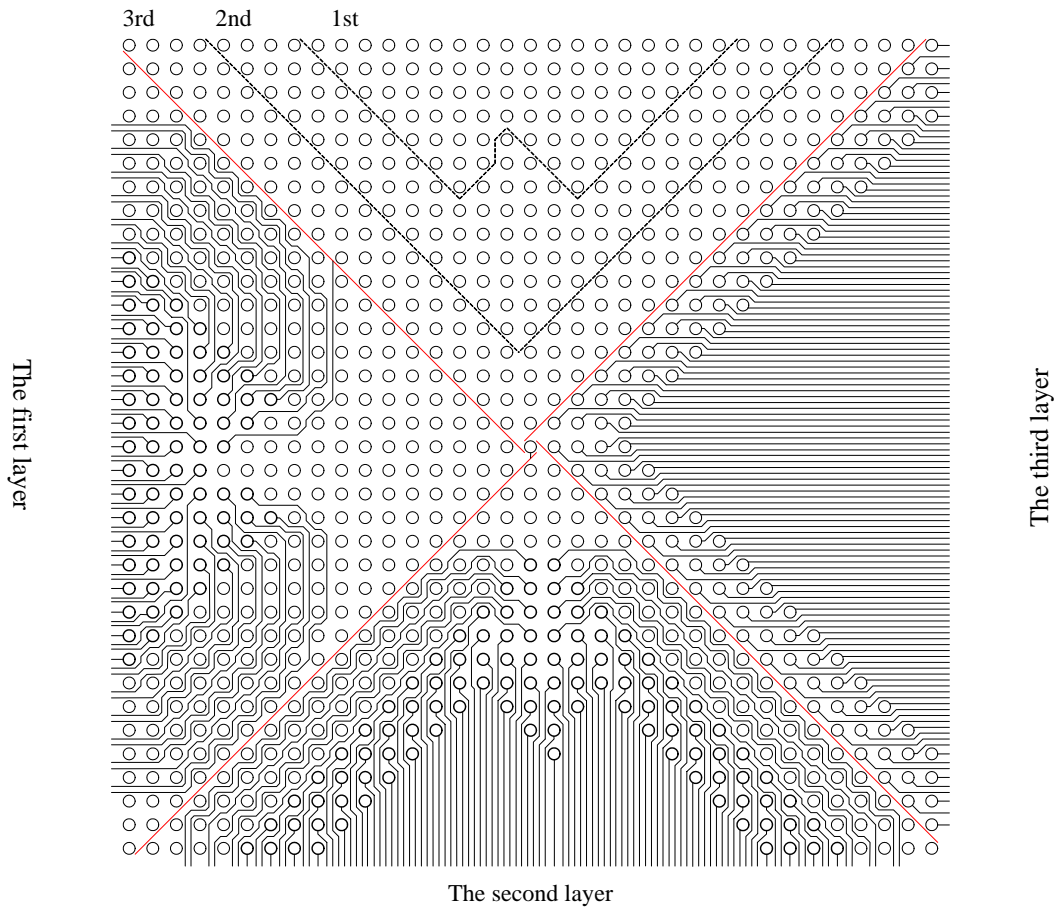
Figure 9. 3-layer escape routing result of a 35×35 FCPBGA case

For high speed circuits, the transmission property of the lines maybe affected by the routing details due to crosstalk effects. In future works, we should add signal integrity constraints into our objective for packaging design of high performance circuits.
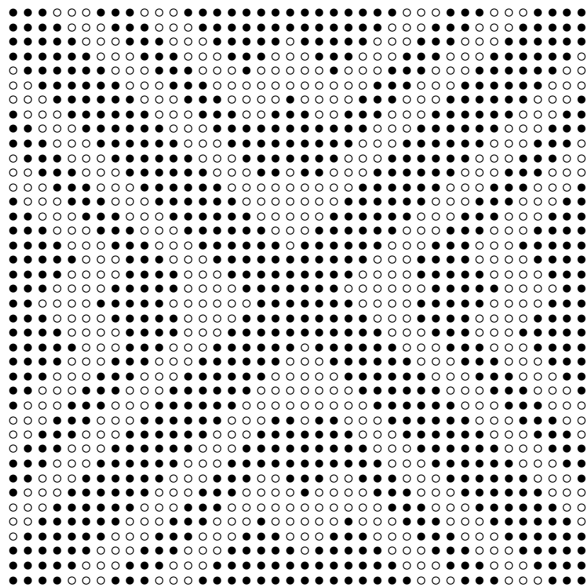

Figure 10. Pin distributions in the 40×40 flip chip - packaging case

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] H. Bakoglu, "Circuits, Interconnections and Packaging in VLSI, Addison-Wesley, 1990.

[2] E. Winkler, "Escape Routing from Chip Scale Packages," IEEE/CPMT Int. Electronics Manufacturing Technology Symp., pp. 393-401, 1996.

[3] M. F. Yu, W. M. Dai, "Single-layer fanout routing and routability analysis for Ball Grid Arrays," IEEE/ACM Int. Conf. on Computer-Aided Design, pp. 581-586, 1995.

[4] N.M. Gasparini, "A Method of Designing a Group of Bumps for C4 Packages to Maximize the Number of Bumps and Minimize the Number of Package Layers," Electronic Components and Technology Conference, pp. 695-696, 1994.

[5] M. Horiuchi, E. Yoda, Y. Takeuchi, "Escape Routing Design to Reduce the Number of Layers in Area Array Packaging," IEEE Trans. on Advanced Packaging, vol.23, no.4, pp. 686-691, Nov. 2000.

[6] W.T. Chan, F.Y. L. Chin, H.F. Ting, "Escaping a grid by edge-disjoint paths," ACM-SIAM Symp. on Discrete Algorithms, pp. 726 - 734, 2000.

[7] R. Shi, H. Chen, C.-K. Cheng, D. Beckman, D. Huang, "Layer Count Reduction for Area Array Escape Routing," Int. Conf. & Exhibition on Device Packaging, Scottsdale, 2005