Efficient Escape Routing for Hexagonal Array of High Density I/Os

Rui Shi
Dept. of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, CA, 92093
rshi@cs.ucsd.edu

Chung-Kuan Cheng
Dept. of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, CA, 92093
kuan@cs.ucsd.edu

ABSTRACT
The chip/package I/Os count has continuously been growing as the systems become more complicated. High density I/Os interconnection and efficient escape routing with high performance and low cost will greatly benefit the whole electronic system. We analyze the properties of the hexagonal array, which can hold about 15% more I/Os compared with the traditional square grid array. We propose three escape routing strategies for the hexagonal array: column-by-column horizontal escape routing, two-sided horizontal/vertical escape routing, and multi-direction hybrid channel escape routing. We can escape I/Os in the hexagonal array in the same or less number of routing layers compared with square grid array. The practical examples show the efficiency of our strategies. Using hexagonal array, we can reduce the number of escape routing layers as well as increase the density of I/Os.

Categories and Subject Descriptors
J.6 [**Computer-Aided Engineering**]: Computer-aided design (CAD); B.7.2 [**Design Aides**]: Placement and routing.

General Terms
Algorithms, Performance, Design.

Keywords
Escape routing, flip chip, BGA, hexagonal array.

1. INTRODUCTION
As the feature size of microelectronic technology becomes smaller, the complexity of electronic systems grows proportionally. 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:

$$N_p = K_p N_g^{\beta} \quad (1.1)$$

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 significantly on the kind of module considered. Today's high-performance ICs exhibit upwards of 2,000 I/O pins, require packages that sometimes exceed 100 layers and will go onto boards with up to 50 layers. New technologies, such as flip chip, CSP, BGA, etc., are developed for chip-level (first-level) and package-to-board (second-level) interconnections to accommodate the increasing demand of high I/O signals count. Area array interconnection is widely used in those advanced technologies. In order to connect the I/O signals in the array to the next level assembly, wires are routed to break I/O signals out which is referred as "escape routing" [8][3][5].

The high I/O signals count and density require an increase in the number of escape routing layers and make the signal integrity issue more serious. An efficient and effective escape routing strategy which achieves high performance with low cost will greatly benefit the electronic product.

Traditionally, the area array is a square grid and I/O signals are escaped row-by-row (column-by-column) from outside. Intuitively some dimensional changes will reduce the number of escape routing layers, such as decreasing wire width and spacing, while these changes expose the system to cost, yield and reliability issues. Gasparini et al. [2] suggested a specific placement of bumps for C4 packages to minimize the package layers count with change in footprint. They sacrificed I/O density to increase wire density and this strategy was not good for the escape routing with multiple layers. Horiuchi et al. [4] proposed a preferential routing strategy, which created specific pad geometry resulting in a high wiring efficiency. This strategy was very practical for assembly of high I/O count flip chip, CSPs, and BGAs. But it was only suitable for the traditional square grid array. Titus et al. [7] presented a "balls shifted as needed" method, which adjusted and optimized I/Os placement in the array to accomplish the escape routing in one layer. They utilized the advantage of hexagonal array but their placement optimization and escape routing strategy were not suitable for multiple layers. In [6], we have discussed the importance of escape sequence for escape routing and proposed two strategies, central triangular and two-sided, which can reduce the number of escape routing layers effectively for square grid array.

In this paper, we analyze the properties of hexagonal array, which increases the density of I/Os in the array remarkably. We propose three escape routing strategies for the hexagonal array: column-by-column horizontal escape routing, two-sided horizontal/vertical escape routing, and multi-direction hybrid channel escape routing. The examples using practical parameters show that our strategies can escape the hexagonal array very efficiently. Thus we can reduce the number of escape routing layers as well as increase the density of I/Os.

The rest of the paper is organized as follows. Section 2 describes the escape routing problem and defines related parameters. The properties of hexagonal array are analyzed in section 3 and

DAC 2006, July 24–28, 2006, San Francisco, California,USA.

detailed comparison between square grid array and hexagonal array is also given. Section 4 explains three escape routing strategies for the hexagonal array. We compare those strategies in section 5 and discuss the conclusions in section 6.

2. PROBLEM DESCRIPTION

In the escape routing problem, I/Os inside the area array are the objects; the corresponding pads are the obstacles and the spaces scattering among the pads in the area array are the routing resources. Fundamentally there are three basic guidelines which are described blow.

(i) Spacing between two wires and the width of the wire. As shown in Figure 1, the wire width is W and the pitch is $(W+S_w)$, where S_w is the spacing between the two consecutive edges of the wires.

(ii) Minimum distance S_p between the edge of the pad and the edge of any metal, as shown in Figure 1.

(iii) Diameter of the pad, D, and pitch, P, between two consecutive pads, as shown in Figure 1.

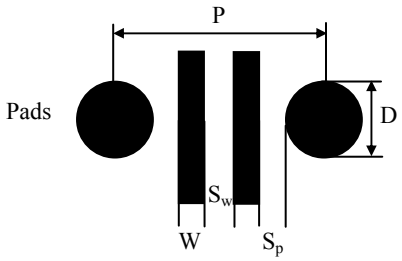

Figure 1. Fundamental parameters

Usually the spacing between two wires S_w and the spacing between pad and wire S_p are identical, we can use S to represent the spacing.

Escape routing breaks out I/Os in the array to the outside and the objective is to minimize the number of routing layers and to break out I/Os as many as possible. In real applications, only signal IOs inside the array require breakout routing. But for analysis, all IOs are taken into consideration.

3. HEXAGONAL ARRAY

The typical I/Os array is a square grid matrix and I/Os are located at the crossing points of the horizontal and vertical mesh. The neighboring four I/Os form a square grid unit. For an n×n square grid array, there are n I/Os in each row and column, n^2 I/Os in the array totally and the area of the array is $(nP+D)^2$. I/Os in the typical square grid array can be shifted and packed further to form a hexagonal pattern. The neighboring six I/Os form a hexagonal unit and one more I/O locates at the center. In a hexagonal array, the angle between the lines joining any adjacent two I/Os is always a multiple of 60°. Figure 2 and Figure 3 show the square grid array and the hexagonal array respectively. We observe that the square grid array is symmetric in horizontal and vertical directions, i.e. if the square grid array is rotated 90° clockwise; it is superposed on the original array. However the hexagonal array doesn't have this properties, it is symmetric in 0°, 60°, and 120° directions, i.e. if the hexagonal array is rotated 60° or 120° clockwise; it is superposed on the original array.

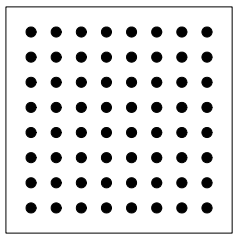

Figure 2. Square grid array

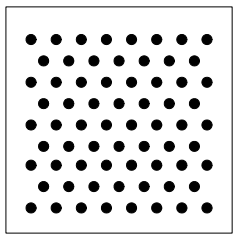

Figure 3. Hexagonal

3.1 Increasing the Number of I/Os

The hexagonal array can hold more I/Os under the same area usage and same I/O pitch constraints, compared with the square grid array. The area of an n×n square grid array is $(nP+D)^2$. Using the same area and same I/O pitch P, the hexagonal array will have n and n-1 I/Os in each row alternately. The number of rows in the hexagonal array is

$$m = \left\lfloor \frac{2}{\sqrt{3}}(n-1) \right\rfloor + 1 \tag{3.1}$$

The number of I/Os in the hexagonal array is

$$TotalNum(hexagon) = \begin{cases} mn - \frac{m}{2} & m = even \\ mn - \frac{m-1}{2} & m = odd \end{cases} \tag{3.2}$$

Plug the formula (3.1) into (3.2), we can prove that the number of I/Os in the hexagonal array is larger than the number of I/Os in the square grid array using same area and same I/O pitch for any n>=8. The increased percentage of the number of I/Os can be approximated as

$$\begin{aligned} &\frac{TotalNum(hexagon) - TotalNum(square)}{TotalNum(square)} \\ &\geq \frac{mn - m/2 - n^2}{n^2} \\ &> \frac{\frac{2}{\sqrt{3}}n^2 - n^2 - \sqrt{3}n - \frac{1}{\sqrt{3}}}{n^2} \\ &= \frac{2-\sqrt{3}}{\sqrt{3}} - \frac{\sqrt{3}}{n} - \frac{1}{\sqrt{3}n^2} \\ &> 15.47\% - \frac{\sqrt{3}}{n} - \frac{1}{\sqrt{3}n^2} \end{aligned}$$

For array with large size, the hexagonal array has more advantage. Table 1 shows their comparison for different size of arrays.

Table 1. Square grid array vs. Hexagonal array using same area, same pitch

Size n	Square grid array		Hexagonal array		Increase
	# rows	# I/Os	# rows	# I/Os	
10	10	100	11	105	5%
15	15	225	17	247	9.78%
20	20	400	22	429	7.25%
25	25	625	28	686	9.76%
30	30	900	34	1003	11.44%
35	35	1225	40	1380	12.65%
40	40	1600	46	1817	13.56%

3.2 Increasing the Spacing between I/Os

The hexagonal array can increase the average I/Os pitch under the similar number of I/Os and same area constraints, compared with square grid array. In an n×n square grid array, the pitch of the adjacent horizontal or vertical I/Os is the minimum pitch P. Holding the same number of I/Os in same area, i.e. n^2 I/Os in area $(nP+D)^2$, the corresponding hexagonal array can separate the I/Os loosely.

We assume the hexagonal array has k and k-1 I/Os in each row alternately, and then according to (3.1) the number of rows in this hexagonal array is

$$l = \left\lfloor \frac{2}{\sqrt{3}}(k-1) \right\rfloor + 1 \tag{3.3}$$

The number of I/Os in the corresponding hexagonal array is

$$TotalNum(hexagon) = \begin{cases} lk - \dfrac{l}{2} & l = even \\ lk - \dfrac{l-1}{2} & l = odd \end{cases} \tag{3.4}$$

Therefore the I/Os pitch P′ in the hexagonal array can be solved from the following inequations

$$\begin{cases} TotalNum(hexagon) \geq n^2 \\ kP' \leq nP \end{cases} \tag{3.5}$$

Plug (3.3) and (3.4) into (3.5), we can prove that the pitch P′ in the hexagonal array is larger than the minimum pitch P for large n. Table 2 shows their comparison for different size of arrays.

Table 2. Square grid array vs. Hexagonal array using same area, holding similar number of I/Os

Square grid array			Hexagonal array			Increase
n	# I/Os	Pitch	k	# I/Os	Pitch	
25	625	P	24	635	1.04P	4.17%
30	900	P	29	941	1.03P	3.45%
35	1225	P	34	1307	1.03P	2.94%
40	1600	P	38	1613	1.05P	5.26%
45	2025	P	43	2083	1.05P	4.65%
50	2500	P	47	2511	1.06P	6.38%

4. ESCAPE ROUTING FOR HEXAGONAL ARRAY

4.1 Column-by-Column Horizontal Escape Routing

The traditional escape routing strategy for the square grid array is to break out the I/Os row-by-row/column-by-column from outside to inside as shown in Figure 4. The spacing between two consecutive I/Os constrains the number of wires going through and limits the number of I/Os escaped for one layer.

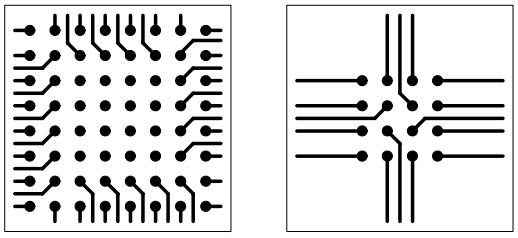

Figure 4. Traditional escape routing for square grid array

The hexagonal array is not symmetric in horizontal and vertical directions. In the compact hexagonal array, in which the pitch of adjacent I/Os is the minimum pitch, the vertical routing channel, as shown in Figure 5, is the edge of the hexagonal unit and the number of wires that can be escaped through is

$$Num(wires)\_verticalChannel = \left\lfloor \frac{P - D - S}{S + W} \right\rfloor \tag{4.1}$$

However the horizontal routing channel, as shown in Figure 5, is a diagonal of the hexagonal unit and has larger capacity:

$$Num(wires)\_horizontalChannel = \left\lfloor \frac{\sqrt{3}P - D - S}{S + W} \right\rfloor \tag{4.2}$$

Therefore the horizontal escape routing can be more efficient than the vertical escape routing. Similarly as the traditional escape routing method for the square grid array, the hexagonal array can be treated as zigzag column array and I/Os can be escaped column-by-column through the horizontal routing channels as shown in Figure 6 and Figure 7.

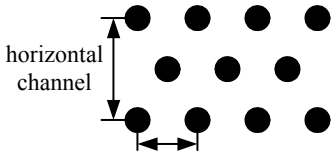

Figure 5. Vertical/horizontal channels

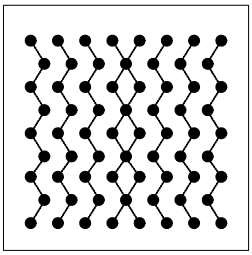

Figure 6. Zigzag columns in hexagonal array

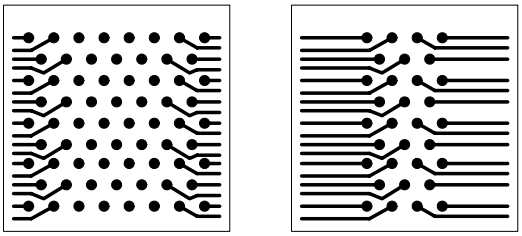

Figure 7. Column-by-Column horizontal escape routing for hexagonal array

Using column-by-column horizontal escape routing method, under some conditions, we can accomplish the escape routing for the hexagonal array with the same number of layers as the square grid array although the hexagonal array holds more I/Os.

We assume the number of wires that can go through the vertical routing channel, i.e. the channel between adjacent I/Os with minimum pitch, is A. Thus for the square grid array, A+1 rows and columns can be escaped in one routing layer and the number of routing layers for breaking out an n×n square grid array using the tradition method is

$$Num(layers)\_square = \left\lceil \frac{n}{2(A+1)} \right\rceil \tag{4.3}$$

The hexagonal array with same area and same pitch also has n zigzag rows. If we assume to use the same number of routing layers as the square grid array, the number of wires that can go through one horizontal channel in hexagonal array should satisfy

$$Num(wires)\_horizontalChannel \geq 2A + 1 \tag{4.4}$$

Combine (4.1) (4.2) with the assumptions, we can derive the condition is

$$\begin{cases} Num(wires)\_verticalChannel = \left\lfloor \frac{P-D-S}{S+W} \right\rfloor = A \\ Num(wires)\_horizontalChannel = \left\lfloor \frac{\sqrt{3}P-D-S}{S+W} \right\rfloor \geq 2A+1 \end{cases}$$

$$\Rightarrow D \geq (2-\sqrt{3})P + W \qquad (4.5)$$

As long as the array's parameters satisfy the condition (4.5), I/Os in the hexagonal array can be escaped within the same number of routing layers as square grid array and the hexagonal array has the same area and same pitch as the square grid array but holds more I/Os. Table 3 shows the practical parameters for flip chip interconnect and fine pitch BGA/CSP, which is provided by ITRS (International Technology Roadmap for Semiconductors) [9]. The condition is usually satisfied.

Table 3. Practical Parameters for condition, unit: μm

	Flip Chip		FBGA/CSP	
Year	**2006**	**2018**	**2006**	**2018**
Pitch	130	70	300	100
Pad Diameter	65	35	120	40
Line Width	27.8	15	36	12
Line Spacing	27.9	15	36	12
Condition (4.5)	✓	✓	✓	✓

We take an 8×8 square grid array as a simple example, as shown in Figure 2. The pitch of every two adjacent horizontal/vertical I/Os is the minimum pitch. This square grid array has 64 I/Os totally. The hexagonal array, which uses same area, can have 68 I/Os as shown in Figure 3 and the pitch of any two adjacent I/Os is also the minimum pitch.

The following values for feature sizes are used in this example and they satisfy the condition (4.5).

1) Minimum pitch (Pmin) = 240μm
2) Diameter of I/O (D) = 110μm
3) Wire width (W) = 43μm
4) Wire spacing (S) = 43μm

The vertical routing channel can route 1 wire and the horizontal routing channel can route 3 wires. Therefore we can use two layers to break out I/Os in that square grid array as shown in Figure 4. We can also use two layers to escape I/Os in the hexagonal array as shown in Figure 7. The number of routing layers for the square grid array and the hexagonal array are identical although the hexagonal array has more I/Os.

4.2 Two-sided Horizontal/Vertical Escape Routing

We have proposed the two-sided escape routing approach for square grid array in [6]. This approach breaks out I/Os in the outside and inside rows/columns simultaneously. It can maintain the outline of the array in a good shape and I/Os are escaped on different routing layers equably. This idea can be exploited in escape routing for hexagonal array to reduce the number of routing layers further.

In the column-by-column horizontal escape routing strategy for the hexagonal array, the number of zigzag columns in the array constrains the number of routing layers. Using the two-sided idea, I/Os in the middle zigzag columns can be escaped through vertical routing channels at the same time as I/Os in the outside zigzag columns are escaped through horizontal routing channels.

We take a 10×11 hexagonal array as an example, which has the same area as a 10×10 square grid array using same pitch. The pitch of any two adjacent I/Os is the minimum pitch and there are 105 I/Os totally. The values for feature sizes are same as the example in section4.1.

Using the column-by-column horizontal escape routing method, we need three layers as shown in Figure 8. However we only use two layers in two-sided horizontal/vertical escape routing approach as shown in Figure 9.

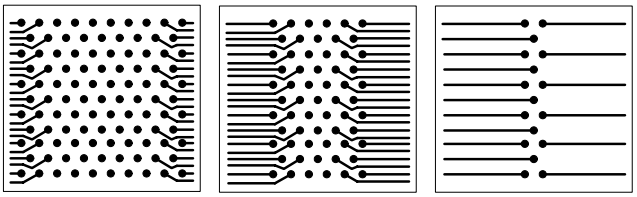

Figure 8. Column-by-column horizontal escape routing

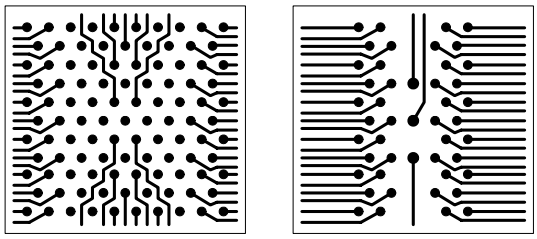

Figure 9. Two-sided horizontal/vertical escape routing

The two-sided approach utilizes the routing channels sufficiently and increases the number of I/Os escaped in every routing layer because it breaks out I/Os inside and outside simultaneously. Compared with the column-by-column horizontal escape routing, it can decrease the number of routing layers efficiently.

4.3 Multi-direction Hybrid Channel Escape Routing

4.3.1 Array Partition and Hybrid Channel

The hexagonal array is symmetric in 0°, 60°, and 120° directions. It can be treated as many nested hexagons as shown in Figure 10. The array can be divided into six partitions and I/Os can be escaped to the outside following six directions. Figure 11 shows this escape routing style.

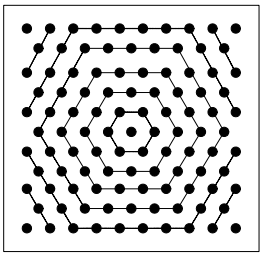

Figure 10. Nested hexagons

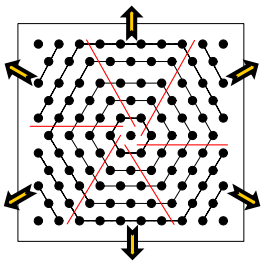

Figure 11. Partition

For each partition, the adjacent I/Os in the same row have the minimum pitch. Instead of breaking out I/Os row by row, we can escaped the I/Os selectively to form indented outline and hybrid routing channels as shown in Figure 12.

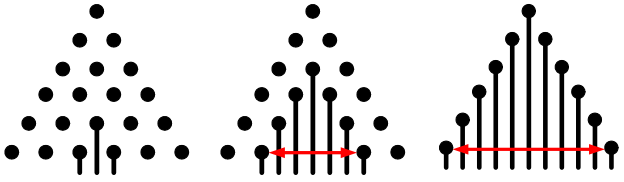

Figure 12. Indented outline and hybrid channel

Because the hybrid routing channel has larger capacity, i.e. it can allow more wires going through, this escape strategy can increase the number of I/Os escaped for one routing layer and potentially reduce the number of routing layers compared with the column-by-column horizontal method. The number of wires going through the hybrid routing channel, which consists of k vertical channels, is

$$Num(wires)\_hybridChannel = \left\lfloor \frac{kP - D - S}{S + W} \right\rfloor \quad (4.6)$$

4.3.2 Escape Routing through Hybrid Channel

Using this multi-direction hybrid channel escape routing strategy, the hexagonal array can be treated as consisting of many indented rows as shown in Figure 13.

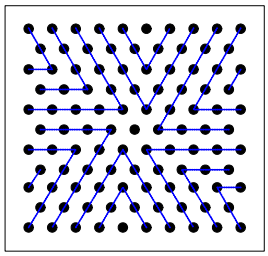

Figure 13. Indented rows in hexagonal array

For an n×m hexagonal array in a square area, which means m and n satisfy the formula (3.1), the number of indented rows in each partition is

$$I \le \left\lfloor \frac{n-2}{3} \right\rfloor + 1 \quad (4.7)$$

We assume the number of wires that can go through the vertical routing channel is A. Thus

$$Num(wires)\_verticalChannel = \left\lfloor \frac{P - D - S}{S + W} \right\rfloor = A \quad (4.8)$$

$$\Rightarrow P - D - S \ge A(S + W)$$

The condition for routing I/Os in A indented rows through the hybrid routing channel, which consists of k vertical channels, is

$$Num(wires)\_hybridChannel = \left\lfloor \frac{kP - D - S}{S + W} \right\rfloor \ge A(2k - 1) \quad (4.9)$$

$$\Rightarrow kP - D - S \ge A(2k - 1)(S + W)$$

Plug (4.8) into (4.9), the condition can be simplified as

$$2(D + S) \ge P \quad (4.10)$$

As shown in Table 4, the condition is generally satisfied.

Table 4. Practical Parameters for condition, unit: μm

	Flip Chip		FBGA/CSP	
Year	**2006**	**2018**	**2006**	**2018**
Pitch	130	70	300	100
Pad Diameter	65	35	120	40
Line Width	27.8	15	36	12
Line Spacing	27.9	15	36	12
Condition (4.10)	✓	✓	✓	✓

Using the multi-direction hybrid channel escape routing method, we break out I/Os at the center of each partition in the first layer to form an indented outline as large as possible and escape at least A indented rows in every following layer as long as the condition (4.10) is satisfied. I/Os in the array can be escaped very efficiently though the hybrid channel.

4.3.3 Automatic Escape Routing Procedure

The hybrid escape routing approach organizes I/Os in the array regularly. The escape routing rules for each partition are identical and the routing for every partition is independent of each other. Furthermore the wires breaking out I/Os go through the hybrid channel orderly. Thus escape routing program can be designed straightforwardly to accomplish this kind of escape routing for any given hexagonal array automatically. The automatic procedure of the multi-direction hybrid channel escape routing is illustrated in Figure 14.

Partition:
Divide the hexagonal array into six partitions.
Labeling:
Identify the indented rows in each partition.
Label I/Os using partition ID and indented rows ID.
Assignment:
For each partition {
Determine the number of indented rows that can be escaped in the first routing layer according to the vertical channel capacity.
Determine the number of indented rows that can be escaped in each following layer according to the hybrid channel capacity.
}
Routing:
For each partition {
Escape I/Os in the first layer symmetrically through vertical channel.
Escape I/Os in each following layer directly through the hybrid channel.
}
Post processing refinement:
Refine the escape routing for I/Os located at special positions, e.g. center, corner, partition boundary, etc., to further reduce the number of routing layers.

Figure 14. Automatic escape routing

4.3.4 Test cases

For the 10×11 hexagonal array example in section4.2, we only need two layers using this multi-direction hybrid channel escape routing strategy as shown in Figure 15.

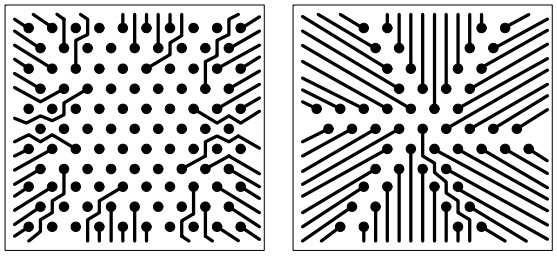

Figure 15. Multi-direction hybrid channel escape routing

5. DISCUSSION

The hexagonal array can hold more I/Os than the traditional square grid array using same area and same pitch, i.e. the hexagonal

array has larger I/Os density. For complicated ICs and packages, the hexagonal array can provide more external interconnection. The approximate increase of I/Os count for large array is 15%. The hexagonal array has larger I/Os' pitch if it uses same area as the square grid array and holds similar number of I/Os. The large pitch will increase the escape routing capacities and can potentially reduce the number of layers for escape routing. Furthermore it can benefit the alleviation of crosstalk.

The column-by-column horizontal escape routing strategy is very straightforward for the hexagonal array. Using this strategy, I/Os in hexagonal array can be escaped in the same number of routing layers as the square grid array which has same area and same pitch although more I/Os are packed inside. However, the vertical routing channels are wasted in this strategy.

The two-sided horizontal/vertical escape routing strategy overcomes the shortcoming of that straightforward strategy. I/Os in the outside zigzag columns are escaped through horizontal channels and simultaneously I/Os in the middle zigzag columns are escaped through vertical routing channels. All the routing channels are utilized sufficiently and the number of escape routing layers is reduced. Nevertheless, there is no simple routing rule for the wires breaking out I/Os in the middle zigzag columns, so it's hard to implement this strategy in automatic program and those wires need to go through many other I/Os, thus crosstalk is an important issue to be considered.

The multi-direction hybrid channel escape routing strategy uses the symmetric property of hexagonal array to divide it into six partitions and exploits hybrid channels to increase the escape efficiency. The hybrid channels increase the number of escape routing wires on every layer and consequentially decrease the number of layers. I/Os in each partition are escaped independently which makes the problem simpler and the wires routing through hybrid channels are very ordered, so this strategy is easy to be implemented in automatic program.

In summary, hexagonal array can be escaped very efficiently as well as providing high density of I/Os.

6. CONCLUTIONS

In this paper, we discuss the advantages of the hexagonal array and propose three escape routing strategies for it. Using same area and same pitch, hexagonal array holds about 15% more I/Os compared with square grid array and our strategies can escape the I/Os in the hexagonal array in the same or less number of routing layers. The practical examples show the efficiency of our escape routing strategies. Using hexagonal array, we can reduce the number of escape routing layers as well as increase the density of I/Os. We continue working on the automatic escape routing program for the hexagonal array and the related crosstalk issues will also be investigated.

7. ACKNOWLEDGMENTS

The authors would like to thank the support from California MICRO program. We would like to thank the reviewers for their constructive suggestions. Thanks are due to Lei Shan and Mark Ritter from IBM for their warm cooperation.

8. REFERENCES

[1] Bakoglu, H.B., *Circuits, Interconnections, and Packaging for VLSI*, Addison-Wesley, Reading, MA, 1990.

[2] Gasparini, N.M., Bhattacharyya, B.K., *A Method of Designing a Group of Bumps for C4 Packages to Maximize the Number of Bumps and Minimize the Number of Package Layers*, In proceedings of 44th Electronic Components and Technology Conference, 1994, 695-699.

[3] Guinn, K.V., Frye, R.C., *Flip-Chip and Chip-Scale I/O Density Requirements and Printed Wiring Board Capabilities*, In proceedings of 47th Electronic Components and Technology Conference, 1997, 649-655.

[4] 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, 686-691.

[5] Katopis, G.A., Zhou, T., *Applications of Wiring Escape Estimation*, In proceedings of 7th IEEE workshop on signal propagation on interconnects, 2003.

[6] Shi, R., Chen, H., Cheng, C.K., Beckman, D., Huang, D., *Layer Count Reduction for Area Array Escape Routing*, In proceedings of IMAPS International Conference and Exhibition on Device Packaging, 2005.

[7] Titus, A., Jaiswal, B., Dishongh, J., and Cartwright, A.N., *Innovative Circuit Board Level Routing Designs for BGA Packages*, IEEE Transactions on Advanced Packaging, vol.27, no.4, Nov. 2004, 630-639.

[8] Winkler, E., *Escape Routing from Chip Scale Packages*, In proceedings of 9th IEEE/CPMT International Electronics Manufacturing Technology Symposium, 1996, 393-401.

[9] *International Technology Roadmap for Semiconductors*, 2003 edition.