Single-Layer Fanout Routing and Routability Analysis for Ball Grid Arrays

Man-fai Yu
Wayne Wei-Ming Dai

UCSC-CRL-95-18
April 25, 1995

Board of Studies in Computer Engineering
University of California, Santa Cruz
Santa Cruz, CA 95064
Phone: +1 (408)459-4954 or +1 (408)459-4234
Fax: +1 (408)459-4829
e-mail: yumf@cse.ucsc.edu or dai@cse.ucsc.edu

ABSTRACT

Fanout routing for Ball Grid Array(BGA) packages becomes non-trivial when the I/O pin count increases. With large number of I/Os, the number of I/Os we can put on a package is not always limited by the available area but sometimes by the ability to fan them out on the next level of interconnect—the PCB or MCM substrate. This paper is the first to consider this problem and offers an efficient algorithm (EVENFANOUT) to solve it. EVENFANOUT generates the optimal uniform distribution of wires. Another important contribution is that we analyzed the relationship between pin pitch and the routability of fanout so that the package designer can choose an optimal pitch for maximum routability. Knowing this relationship, we know whether a fanout routing is routable or not before it is routed. This is implemented in the Package Early Analysis and Routing Tool (PEART) for rapid development of Ball Grid Array Packages.

Keywords: package routing, ball grid array, pin grid array, planar routing, routability, even wiring, fanout routing

1 Introduction

Increasing demand for I/O pin-count prompts the packaging industry to look for more space-efficient packaging methods. Ball-grid arrays (BGAs) promise more I/Os in less area. However, some pins (or solder bumps) may not be available for signal I/O because there is not enough room on the substrate to fan the pins out for connection. Therefore the number of pins we can put on the package may be limited by the routability of the fanout routing on the substrate or PCB. Figure 1 shows a BGA package with its fanout routing on the substrate. The routing may be so dense that even though there is room for more balls on the package, there may not have enough routing space to route to them.

Since a fully populated package may not have a routable fanout routing, the package designer has to know how many pins we can put on a package before the fanout routing becomes unroutable. The package designer can trade off pin pitch against the number of rings. For a fixed number of total pin count, fewer rings cause smaller pin pitches and vice versa. So it is not intuitively clear

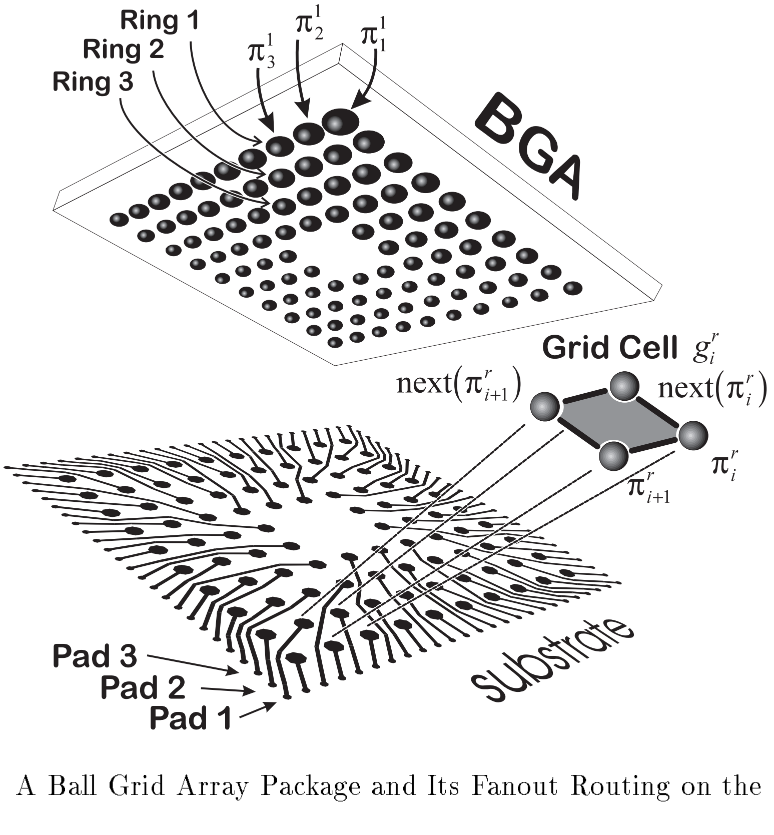

Figure 1: A Ball Grid Array Package and Its Fanout Routing on the Substrate

how to trade off pin pitch and the number of rings to get the most routable fanout.

In a similar case for area I/O chips, the chip size must be at least as large as the size of the I/O array on the chip which in turn is determined by the routability of the fanout. This is an important limitation as chips shrink. The question is: given the number of I/Os, what is the smallest chip size for a routable fanout?

To answer these questions, we must be able to determine whether a package is routable or not. A package is routable if there exists a routable topological fanout routing. There is no netlist related to a fanout routing because the exact position where a pin is to be routed is not fixed. The only requirement is that all pins must be routed to the periphery of the array.

In this paper we describe an algorithm that generates a topological routing for a fanout. The algorithm has the most uniform distribution of wires to maximize routability. We also find the set of critical cuts and a tight lower bound on the density of these cuts as a closed form. Hence we know a package is routable (or not) before it is routed. This is implemented in the Package Early Analysis and Routing Tool (PEART) for designing BGAs.

Darnauer and Dai[1] proposed a similar algorithm, although under a different context. They have shown that their algorithm generates a routing whose critical density is no more than $\sqrt{2}$ times the density of the perimeter cuts. Since our algorithm creates the same topological routing, it also has this property. However, we believe that our algorithm is simpler and can be extended to irregular shapes more easily. More importantly, we have found the set of critical cuts and a tight lower bound of its density. This information is more useful because it allows us to determine the routability of a routing without actually route it.

2 Problem Definition

Since the fanout problem applies to BGAs, PGAs and area I/O chips, we have to generalize some terms. We define *pins* to be connectors on the package that are arranged in a grid array. They may be solder bumps on a BGA or a chip. *Pads* are via pads outside the pin array which are just a set of destinations for the wires. Assume a BGA package has T pins. Let the set of pads $X = \{1, 2, \ldots, T\}$ be arranged in a clockwise manner starting at an arbitrary corner (see Figure 1). The package also has R pin rings $\Pi = \cup_{r=1}^{R} \Pi^r$, where each ring r consists of P_r pins $\Pi^r = \{\pi_1^r, \pi_2^r, \ldots, \pi_{P_r}^r\}$. Each

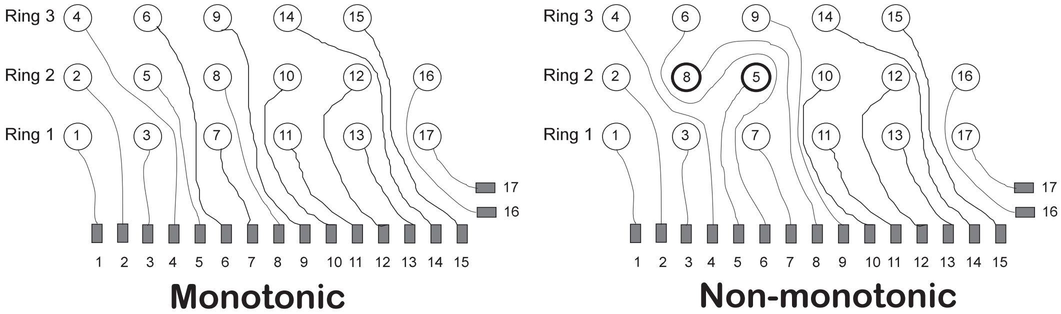

Figure 2: Monotonic vs. non-monotonic topological routing

ring starts at the same corner as the pads. Since the pins are arranged in rings, all arithmetic involving the subscript of a pin should be modulo P_r. To simplify the mathematics, we assume that the number of pins of the outermost ring, ring 1, is divisible by 8. If the number of pins in ring 1 is $8N$, we have

$$T = 4R(2N - R + 1),$$

and

$$P_r = 8(N - R + 1).$$

The problem of creating a fanout on a single-layer for a BGA (1LFANOUT) can be defined as follows:

Problem 1 (1LFANOUT): *The* single-layer fanout routing (1LFANOUT) *problem is to create detailed routing given the set of pins* Π *and the set of pads* X *such that each pad is connected to one and only one pin.*

1LFANOUT assumes that the number of pads and pins are the same so that connections has to be made for all pins, i.e. $|\Pi| = |X|$. Since all pads are equivalent, we have the freedom of choosing which pad to route for a pin. A *pin assignment* is a one-to-one and onto mapping $\Phi : \Pi \rightarrow X$. There are many topological routings for a given instance of 1LFANOUT. We are particularly interested in those that have no detours. Formally, we define *monotonic topological routing (MTR)* for fanout routing as follows:

Definition 1: *A* monotonic topological routing (MTR) *is a topological routing such that a* $w = (\pi_i^r, p)$ *connecting* π_i^r *and pad* p *intersect at most one cut* (π_k^s, π_{k+1}^s) *for some* k *in each ring* s*.*

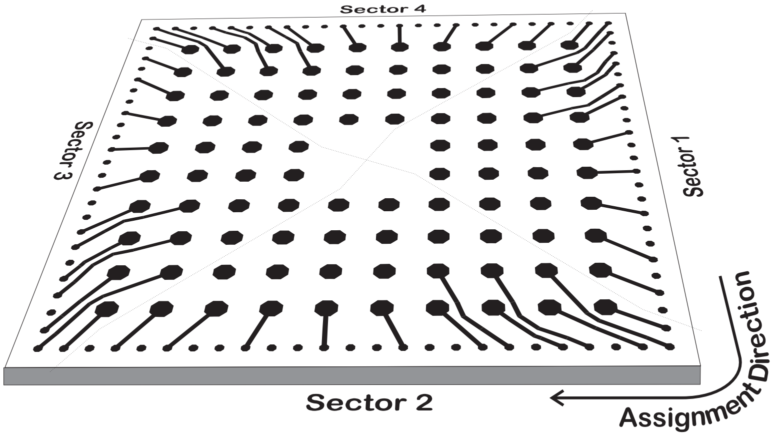

Figure 3: Pin Assignments in Each Sector of a Package

This definition is similar to the MTR definition for PGA routing[2]. Intuitively, MTR specifies a class of topological routing where all wires originate from the pads enter the pin array without any detours. Once a wire enters a ring, it never leaves. Figure 2 shows a simple example of an MTR and a non-monotonic topological routing. Note that the assignment order on ring 2 is violated in the non-monotonic routing. Yu and Dai[2] showed that an MTR has a unique pin assignment known as the *monotonic pin assignment (MPA)* where MPA is formally defined as follows:

Definition 2: *A* monotonic pin assignment (MPA) *is a pin assignment such that for all* r*,* $\Phi(\pi_i^r) > \Phi(\pi_j^r)$ *if and only if* $i > j$*.*

Therefore the problem of creating an MTR is reduced to creating its corresponding monotonic pin assignment.

3 The EVENFANOUT Algorithm

In this section we present an algorithm that generates a "good" monotonic pin assignment so that the wires are distributed as uniformly as possible. Evenly distributed wiring is highly desired in a package design due to routability, performance and technology concerns. An evenly distributed routing is the least congested and the average wire-to-wire distance is also the highest, which means that crosstalk level and yield are both optimized.

The EVENFANOUT algorithm (Figure 4) creates a monotonic pin assignment on a ring-by-ring basis starting from the outermost ring. It takes advantage of the symmetric geometry of the

Algorithm 1 (EVENFANOUT):

```
Algorithm EVENFANOUT(Set of Pins $\Pi$, Ordered set of pads $X$)
    For rings $r \leftarrow 1$ to $R$
        $k \leftarrow \lfloor |X|/P_r \rfloor$                    ...(*)
        $q \leftarrow |X| - kP_r$
        $t \leftarrow (P_r - q)/2$
        $j \leftarrow 1$
        For pads $i \leftarrow 1$ to $|X|/4$
            ASSIGN4($X[i], j, r$)
            $j \leftarrow j + 1$
            If $i \geq tk$ and $i \leq tk + q(k+1)$
            Then $i \leftarrow i + k + 1$
            Else $i \leftarrow i + k$
        Endfor
        Remove all assigned pads from $X$
    Endfor

Subroutine ASSIGN4($i, j, r$)
    Assign pad $i$ to pin $\pi^r_j$                              (Sector 1)
    Assign pad $|X|/4 - i + 1$ to pin $\pi^r_{P_r/4-j+1}$        (Sector 2)
    Assign pad $|X|/2 - i + 1$ to pin $\pi^r_{P_r/2-j+1}$        (Sector 3)
    Assign pad $3|X|/4 - i + 1$ to pin $\pi^r_{3P_r/4-j+1}$      (Sector 4)
```

Figure 4: Algorithm EVENFANOUT

package and divides it into four sectors. Figure 3 shows the dividing lines. The algorithm assigns pins in the clockwise direction. For each ring, it computes a basic step size k and assign every kth or $(k+1)$st pad to the next pin in the ring. After assigning one ring, the assigned pads are removed from X and the process repeats for the next ring.

Since each pad is visited once, EVENFANOUT runs in order $O(|X|)$. The storage complexity is the size of the output, i.e. $O(|\Phi|) = O(|X|)$.

Also note that all arithmetics are integer. All divisions have no remainders except the line marked (*). We can observe the following invariants:

- The number of pads assigned at the end of i-loop is P_r.
- $|X|$ is divisible by 8.
- q is divisible by 8.

It is straightforward to verify that EVENFANOUT creates an MPA. On every ring, the pins

Algorithm 2 (MAKEMTR):

```
Algorithm MAKEMTR($\Phi, X, \Pi$)
    For $r \leftarrow 1$ to $R$
        For $i \leftarrow 1$ to $P_r$
            Route $\pi_i^r$ to $\Phi(\pi_i^r)$
```

Figure 5: Algorithm MAKEMTR

are assigned in increasing order. The pads are assigned in the same ordering as the pins. The four assignments in ASSIGN4 do not cross each other because they falls into disjoint sectors.

After the assignment, we can generate the topological routing using the simple algorithm MAKEMTR (Figure 5).

The routing from π_i^r to $\Phi(\pi_i^r)$ is created by the shortest-path algorithm described by Dai, Dayan and Staepelaere[3]. After the topological routing is created, design rules can be enforced as proposed by Dai, Kong and Sato[4] which is based on Maley's[5] routability test using the rubber-band sketch model. The rubber-band representation of topological routing of Surf is described by Dai, Kong, Jue and Sato[6].

The time complexity of MAKEMTR is $O(|X|S)$ where S is the complexity of creating rubber-band routing for a single wire, i.e. the complexity of the Route routine.

4 Uniform Wiring Distribution

An important property of EVENFANOUT is that it distributes the wires as uniformly as possible around the rings. The pin assignment ordered the wiring in such a way that no crossing is necessary.

To simplify the mathematics, we assume that the pin pitch is unit distance. Let the expression next(π_i^r) denote the pin on ring $r+1$ that is closest to π_i^r. Similarly, prev(π_i^r) denote the pin on ring $r-1$ that is closest to π_i^r. Note that prev() is undefined for a corner pin. At a corner, three pins have the same next()—the corner pin and its two neighbors on the same ring. We define a *grid cell* g_i^r to be the area bounded by the pins π_i^r, π_{i+1}^r, next(π_i^r) and next(π_{i+1}^r). Pin π_i^r cannot be a corner pin. If π_{i+1}^r is a corner pin, the cell is defined by the area bounded by π_i^r, π_{i+1}^r, π_{i+2}^r and next(π_i^r). Intuitively, π_i^r is at the lower right corner of the cell. We define the *bottom cut* of a cell g_i^r to be the cut (π_i^r, π_{i+1}^r), the *top cut* to be (next(π_i^r), next(π_{i+1}^r)), the *left side cut* to be (next(π_{i+1}^r), π_{i+1}^r), the *right side cut* to be $(\text{next}(\pi_i^r), \pi_i^r)$. The left and right side cuts are collectively known as the side cuts of a cell. Figure 1 shows a pin and its related grid cell.

Now we proceed to prove that EVENFANOUT produces a uniformly distributed topological routing. The following lemma states that the wires are uniformly distributed among the cuts in all the rings. More precisely, it says that the number of wires between two adjacent pins in the same ring differs at most by one.

Lemma 1: *For all $r = 1, \ldots, R$,*

$$\max_{\forall i}(F(\pi_i^r, \pi_{i+1}^r)) - \min_{\forall i}(F(\pi_i^r, \pi_{i+1}^r)) \leq 1.$$

$F(\pi_i^r, \pi_{i+1}^r)$ is the flow of the cut (π_i^r, π_{i+1}^r), i.e., the number of wires intersecting the cut.

Proof: In ring r, i is incremented by either k or $k+1$ in each iteration. Hence the number of wires between any two adjacent pin in ring r is either $k-1$ or k. Therefore

$$\max_{\forall i}(F(\pi_i^r, \pi_{i+1}^r)) - \min_{\forall i}(F(\pi_i^r, \pi_{i+1}^r)) \leq k - (k-1) = 1.$$

□

We will now use the result of Lemma 1 to derive a useful relationship between k_r and k_{r+1} where k_r is the value of k in EVENFANOUT in the rth iteration. The following lemma can be derived easily from the definition of k_r.

Lemma 2: $0 \leq k_r - k_{r+1} \leq 1$ *for all r and* $k_r - k_{r+2} \geq 1$.

Proof: Let X_r be the set X at iteration r. At iteration r, $|X| = T - 4(r-1)(N-(r-1)+1) = T - 4(r-1)(N-r)$. From EVENFANOUT,

$$k_r = \lfloor |X_r|/P_r \rfloor, k_{r+1} = \lfloor |X_{r+1}|/P_{r+1} \rfloor.$$

$P_{r+1} = P_r - 8 = 8(N-r)$. Substituting in all variables, we have,

$$k_r - k_{r+1} \quad \geq \quad \frac{|X_r|}{P_r} - \frac{|X_{r+1}|}{P_{r+1}} - 1$$

$$= \frac{-T + 4(r-1)(2N-r+2) + 8(N-r+1)}{8(N-r+1)(N-r)} > -1$$

since $1 \leq R \leq N$.

$$\begin{aligned} k_r - k_{r+1} - 2 &\leq \frac{|X_r|}{P_r} - \frac{|X_{r+1}|}{P_{r+1}} - 1 \\ &= \frac{-|X_r| - 8(N-r+1)^2}{8(N-r+1)(N-r)} < 0. \end{aligned}$$

Combining the two results we have the lemma. □

In each iteration of the i-loop, EVENFANOUT choose to use either k_r or $k_r + 1$ as the step size. Since Lemma 2 states that $k_r - k_{r+1} \leq 1$, it is possible that the difference of step sizes between adjacent rings be -1, 0, 1 or 2. The following lemma states that this difference can only be either 0 or 1 due to the behavior of the remainders q_r and q_{r+1}.

Lemma 3: *If $k_{r+1} = k_r$, $q_r - q_{r+1} = P_r - 8k_r \geq 0$. If $k_{r+1} = k_r - 1$, $q_r - q_{r+1} = -8(k_r - 1) \leq 0$.*

Proof: If $k_{r+1} = k_r - 1$,

$$\begin{aligned} q_r &= |X_r| - k_r P_r, \\ q_{r+1} &= |X_{r+1}| - k_{r+1} P_{r+1} \\ &= |X_r| - P_r - (k_r - 1)(P_r - 8) \end{aligned}$$

Combining the two, we have

$$q_{r+1} - q_r = 8k_r - 8 \geq 0$$

since $k_r \geq 1$.

Similarly, when $k_{r+1} = k_r$,

$$\begin{aligned} q_r &= |X_r| - k_r P_r, \\ q_{r+1} &= |X_{r+1}| - k_{r+1} P_{r+1} \\ &= |X_r| - P_r - k_r(P_r - 8) \end{aligned}$$

Combining,

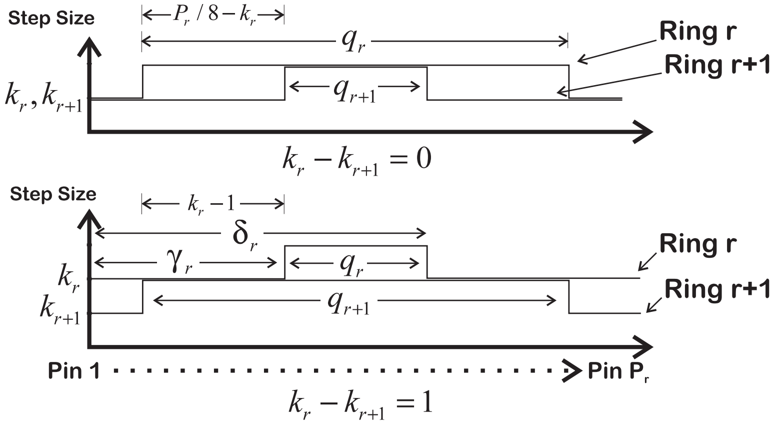

Figure 6: Different flows between adjacent rings

$$q_r - q_{r+1} = P_r - 8k_r$$

We can show that $P_r - 8k_r \geq 0$ by induction. Consider

$$\begin{aligned} P_1 - 8k_1 &= 8N - 8\lfloor\frac{4R(2N-R+1)}{8N}\rfloor \\ &\geq 8N - 8\frac{4R(2N-R+1)}{8N} \\ &= 4(2N(N-R)+R(R-1))/N \geq 0 \end{aligned}$$

since $1 \leq R \leq N$. Now consider ring $r+1$. Since the pads assigned to the first r rings are removed from X, this is exactly the problem instance (Π', X') where $\Pi' = \cup_{i=r+1}^{R}\Pi^i$ and $X' = X -$ all assigned pads. By a similar argument as ring 1, it is true for ring $r+1$. □

With the above lemma, we can show that the difference of steps between two rings differs at most 1 in a grid cell.

Lemma 4: *If T_i^r is the flow of the top cut of a cell g_i^r and B_i^r is the flow of the bottom cut of the same cell,* $0 \leq B_i^r - T_i^r \leq 1$.

Proof: Figure 6 shows the relationship of the quotients q_r and q_{r+1} in the two cases where $k_r = k_{r+1}$ and $k_r - 1 = k_{r+1}$. Now consider the case $k_r = k_{r+1}$. We can divide the i-loop into 5 regions.

Region I $j \leq \gamma_r$. In this region $T_i^r = k_{r+1} - 1 = k_r = B_i^r$ so $B_i^r - T_i^r = 0$.

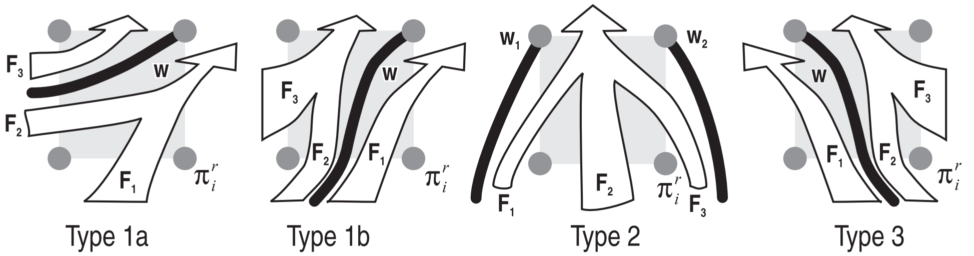

Figure 7: The flows inside a grid cell

Region II $\gamma_r < j \leq \gamma_{r+1}$. In this region $T_i^r = k_r - 1$ and $B_i^r = k_r$ so $B_i^r - T_i^r = 1$.

Region III $\gamma_{r+1} < j \leq \delta_{r+1}$. In this region $T_i^r = k_{r+1} = k_r = B_i^r$ so $B_i^r - T_i^r = 0$.

Region IV and V These regions are the same as Region II and I respectively due to symmetry. Therefore in any case $1 \geq B_i^r - T_i^r \geq 0$. The analysis for $k_r - 1 = k_{r+1}$ is similar. □

From Lemma 4, we can find the difference of neighboring side cuts. Let the flow through the left and right side cut of the grid cell g_i^r be L_i^r and R_i^r respectively. If the number of pins connected to a wire in the cell is a, we have the following equation.

$$L_i^r - R_i^r + a = B_i^r - T_i^r \tag{4.1}$$

Figure 7 shows the three types of cells based on how the pins in the cell is connected. The first cell (rightmost cell) in ring r is always Type 3. In this cell, $L_1^r = B_1^r = k_r - 2$. Now consider its neighboring cells. If cell g_i^r is Type 3, we have $L_{i+1}^r - R_{i+1}^r = B_{i+1}^r - T_{i+1}^r - 1$ since $a = 1$ in g_i^r. Since $B_i^r - T_i^r \leq 1$ by Lemma 4, $R_i^r \geq L_i^r \geq 0$. Hence, g_{i+1}^r is Type 3.

From Lemma 3, the number of cells where $B_i^r - T_i^r = 0$ is always $8k_r - 8$ per ring (Figure 6). Since $R_1^r = k_r - 2$, there are at least $k_r - 2$ Type 3 cells in which $B_i^r - T_i^r = 0$ before a Type 3 cell g_i^r with $L_i^r = 0$ exists. Therefore the next cell g_{i+1}^r must be Type 2 because $B_{i+1}^r - T_{i+1}^r = 0$ in this cell.

The cell next to a Type 2 cell is always a Type 1b cell because L_i^r is always less than or equal to R_i^r and the flow of the left side cut of a Type 2 cell, L_i^r, is 0. When $L_i^r = k_r - 1$, the next cell is Type 1a.

Since the difference of side flows only change when $B_i^r - T_i^r = 0$ and there are only $2(k_r - 1)$ in

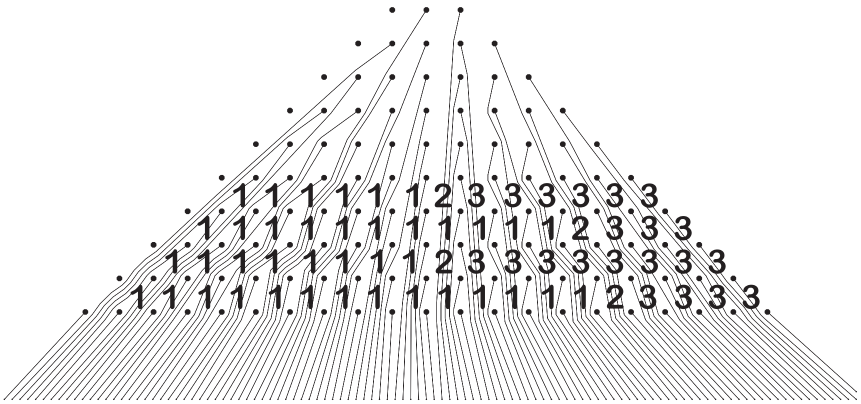

Figure 8: Cells in sector 2 and their types

a sector, the maximum side flow of any cell is at most $k_r - 1$ which is at the left boundary of the sector on the ring. Figure 8 shows the cells of a sector.

We summarize the result into the following theorem.

Theorem 1: *The flow of any cut in a grid cell in ring r is less than or equal to $k_r - 1$ and the side flow changes monotonically with the difference of at most 1 along the ring.*

5 Routability Analysis

In this section we present the set of critical cuts for any package routed with EVENFANOUT. If all of these cuts does not overflow, we know that the package is routable. We also know the density of all the cuts in the set.

The critical cut is the densest cut of the whole design. The density of a cut is defined by the quotient of the flow of the cut (the number of wires intersecting the cut) and the capacity of the cut (the Euclidean length of the cut). A design may have more than one critical cuts where they all have the same density. Maley[5] showed that we only need to check the cut that is the closest between two obstacles which in our case are the pins. In general we need to check $O(G^2)$ cuts under the Euclidean wiring metric where G is the number of obstacles. However, due to the highly regular and symmetric configuration of the package, we can find the set of critical cuts for topological routing generated by EVENFANOUT easily and in fact in most cases we only need to check a few cuts.

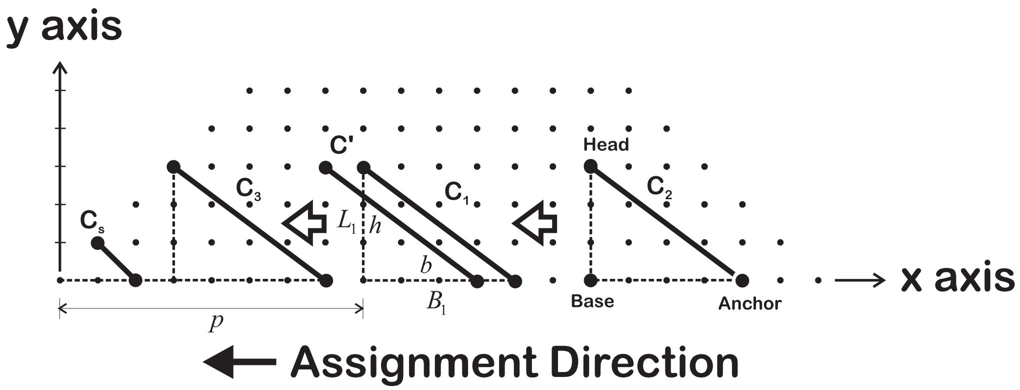

Figure 9: Cuts within a sector.

In the following we progressively compare classes of cuts and eliminate those of less dense.

The previous coordinate system for pins are very useful in EVENFANOUT. However, it is cumbersome to refer pins on different rings. We will adopt the grid coordinate for further analysis. We place the origin on the lower left corner of Sector 2 (Figure 9) and X and Y axis in the usual direction. Since the pin pitch is normalized to unit distance, each pin has an integer coordinate. Under this new coordinate, we denote pin ρ by (ρ_x, ρ_y).

For a cut $C(\rho, \sigma)$, $\rho_y < \sigma_y$ and $\rho_x > \sigma_x$, we denote head(C) be σ, anchor(C) be ρ and base(C) be the pin $\tau = (\sigma_x, \rho_y)$. The *cut triangle* of a cut is the triangle bounded by its head, anchor and base.

5.1 Cuts within a Sector

First we consider the cuts within a sector. Figure 9 shows that there are 3 types of cuts. Since the sector is symmetric, we only need to consider cuts with negative slopes. We will show that for a given length and slope, the cut that ends at the right and bottom boundaries, i.e. cuts like C_3 have the highest density.

We can eliminate cuts like C_2 very easily. Notice that $\Phi(\text{base}(C_2)) \leq \Phi(\text{head}(C_2)) \leq \Phi(\text{anchor}(C_2))$. This means that some wires entering the triangle from the bottom leaves between head(C_2) and base(C_2). We can "shift" this cut toward the center of the sector horizontally and reduce this flow. Hence there exists a denser cut to the left of C_2.

Since EVENFANOUT solves a ring and reduce the problem to a subproblem with one less ring, it is sufficient to consider cuts with its base and anchor on ring 1. For a given cut C_1, we look at the cut one grid to its left, C'. C' is usually denser than C_1 so we can "move" a cut to the left.

Consider the case $k_1 = k_2$. When $p > \gamma_1$, the flow $B_1 = b(k_1 + 1)$ is the same between C' and C_1. The flow L_1 increased because the side flows of grid cells increase monotonically along the assignment direction (Theorem 1). Therefore C' is denser or equally dense than the original cut C_1.

If $p \leq \gamma_1$, the flow across the base of the triangle of the new cut C' is one less than B_1. This is because the move causes the flow of the bottom cut of a grid cell changes from k_1 to $k_1 - 1$. However, consider the grid cell of the base pin of C', g. The top and bottom flow is the same for this cell (Figure 6). Therefore by Lemma 4, the difference of the left side cut and the right differs by 1. Therefore the flow across the left side of the cut triangle of C', L'_1 is at least one greater than L_1. Therefore the density of the new cut C' is greater than or equal to the original cut C_1.

We can repeat the above argument and move the cut horizontally towards the left corner. The process stops when head(C_1) is at the boundary (C_3 in Figure 9).

For a cut that touches the boundary, we can compute the upper bound of its density. The flow across the cut, F is $B + L - V$ where B is the flow that cross the bottom of the triangle, L is the flow across the left side and V is the number of connections within the triangle. By carefully considering the rounding effect, we can obtain the following expression:

$$V = \lfloor (b+1)(h+1)/2 \rfloor - 1$$

where b and h is the base and height of the triangle respectively. L is the sum of side flows throughout the left side of the triangle. It is less than or equal to the sum of maximum side flow on each ring, i.e. $L \leq \sum_{i=1}^{h} k_i$. We use k_i instead of $k_i - 1$ as stated in Theorem 1 because the pins on the side is also counted as connected within the triangle. Combining all the expressions, we have

$$\begin{aligned} D(C_3) &\leq \frac{b(k_1+1) + \sum_{i=1}^{h} k_i - \lfloor (b+1)(h+1)/2 \rfloor + 1}{\sqrt{b^2 + h^2}} \\ &\leq ((b+h)(k_1+1) - (b+1)(h+1)/2 - 1/2)/\sqrt{b^2 + h^2} \end{aligned}$$

We can maximize the above expression by substituting $b = W\cos\theta$ and $h = W\sin\theta$ with the constraint $b > 0$ and $h > 0$. The expression is maximum when $\theta = \pi/4$. We have

$$D(C_3) \leq \sqrt{2}(k_1 + 1/2) - W/4 + 1/2W$$

This is the maximum when W is the minimum at $\sqrt{2}$ which corresponds to $b = h = 1$, i.e. the cut C_s. Hence we conclude that

Lemma 5: *If $k_1 = k_2$, the densest cut is C_s with density $\sqrt{2}(k_1 - 1)$.*

Now consider the case $k_1 = k_2 + 1$. When $p > \gamma_2$, the difference of side flows of the grid cell of head(C_1) is 1. So $L_1' \geq L_1 + 1$. The bottom flow B_1' may be one less than B_1 because the bottom flow of a grid cell in B_1 changes from k_1 to $k_1 - 1$. Hence the density of the new cut $D(C') \geq D(C_1)$.

We can again repeat the argument and "move" the cut toward the left corner. The argument no longer applies when $p = \gamma_2$. We now compute an upper bound for cuts whose head and base is at γ_2.

The approach is the same as in case $k_1 = k_2$ and it turns out that the expression is exactly the same. This is because we does not use the fact that the head of the cut is at the boundary during our computation in case $k_1 = k_2$. Hence the most critical cut is the diagonal cut of the grid cell g at γ_2 of ring 1. However, the density of this cut is the same as the cut C_s because the side flow and bottom flow are the same along the string of cells between g and the corner cell. The side flow does not change from one cell to the next because the difference between the top flow T_i^r and the bottom flow B_i^r of these cells is 1 (Figure 6). Hence, again we conclude that the densest cut is C_s in this case. Combining the previous case, we have the following conclusion.

Lemma 6: *The densest cut in the sector is the diagonal cut C_s at the lower left corner of the sector, if $k_1 \geq 4$, which has a density of $\sqrt{2}(k_1 - 1)$. Otherwise the densest cut is the bottom cut of the cells g_i^1 where $\gamma_1 < i \leq \delta_1$. Their densities are k_1.*

The $k_1 \geq 4$ condition comes from the consideration of the bottom cut of the cells at the center of the sector on ring 1. The density of these cuts can be as large as $k_1/1 = k_1$. When $k_1 \geq 2 + \sqrt{2}$, the diagonal cut C_s dominates.

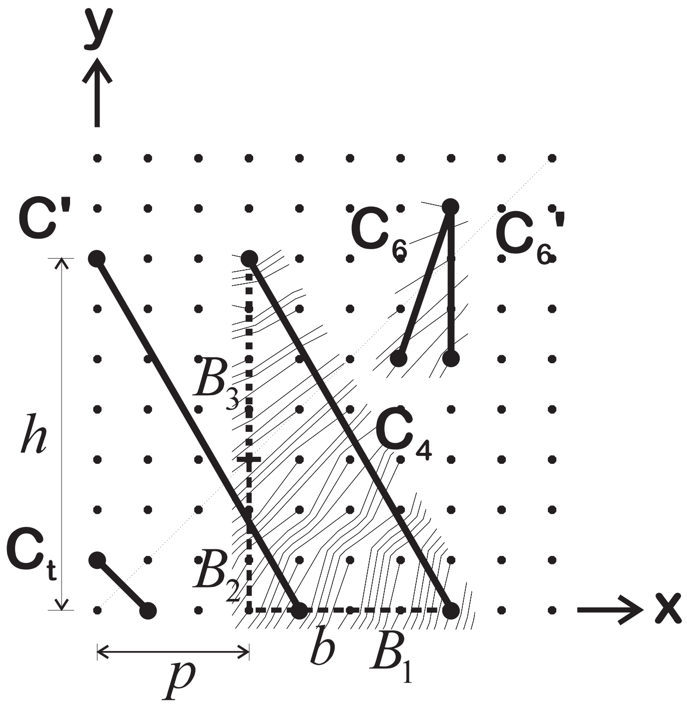

Figure 10: Cuts between two sectors

5.2 Cuts between two sectors

In this subsection we investigate cuts across a sector boundary. We will show that the densest cut is C_t in Figure 10. We can immediately dismiss cuts with positive slopes like C_6 because $\Phi(\mathrm{base}(C_6)) > \Phi(\mathrm{anchor}(C_6)) > \Phi(\mathrm{head}(C_6))$. Some wires enter the bottom of the triangle and leave on the right side without intersecting the cut. The cut C_6' is both shorter and captures these flows so it is denser than C_6. Therefore we only need to consider cuts with negative slopes.

The flow across C is the number wires across B_1, B_2 and B_3 minus the number of connected pins inside the area bounded by the dotted lines, V.

When $k_1 = k_2 + 1$, we can show that the density of C' is greater than or equal to C, i.e., the cut that ends at ring 1 on both ends of the same slope is denser. For C, We have

$$\begin{aligned} B_1 &= bk_1 + (b+p-\gamma_1)\,u(b+p-\gamma_1) - (b+p-\delta_1)\,u(b+p-\delta_1) \\ B_2 &\leq \sum_{i=1}^{p} k_i \end{aligned}$$

$$\begin{aligned} B_3 &= (h-p)k_{p+1} + (h-p-\gamma_{p+1})\,u(h-p-\gamma_{p+1}) - (h-p-\delta_{p+1})\,u(h-p-\delta_{p+1}) \\ V &= \lfloor (b+1)(h+1)/2 \rfloor - 1. \end{aligned}$$

For C', $B_1' = bk_1 + (b-\gamma_1)\,u(b-\gamma_1) - (b-\delta_1)\,u(b-\delta_1)$, $B_2' = 0$, $B_3' = hk_1 + (h-\gamma_1)\,u(h-\gamma_1) - (h-\delta_1)\,u(h-\delta_1)$ and $V' = V$.

Since

$$\begin{aligned} B_1' - B_1 &\geq -p \\ B_3' - B_2 - B_3 &\geq hk_1 - p(k_1-1) - (h-p)(k_1-1+1) = p, \end{aligned}$$

we have $F(C') - F(C) = B_1' + B_3' - V' - B_1 - B_2 - B_3 + V \geq 0$. Hence we have the following lemma.

Lemma 7: *If $k_1 = k_2 + 1$, a cut is less than or equal to the cut that ends on ring 1 with the same slope.*

Now,

$$\begin{aligned} D(C') &\leq \frac{b(k_1+1) - 1 + h(k_1+1) - 1 - \lfloor (b+1)(h+1)/2 \rfloor + 1}{\sqrt{b^2+h^2}} \\ &\leq \frac{(b+h)(k_1+1) - (b+1)(h+1)/2 - 1/2}{\sqrt{b^2+h^2}} \\ &\leq (k_1+1/2)(\cos\theta + \sin\theta) - \frac{W}{4}\sin 2\theta - 1/R. \end{aligned}$$

where we substitute $b = W\sin\theta$ and $h = W\cos\theta$. The above maximizes at $\theta = \pi/4$ and $R = \sqrt{2}$ under the constraint $b > 0$ and $h > 0$ so we have the following lemma.

Lemma 8: *If $k_1 = k_2 + 1$, the density of any cut is less than or equal to the density of the cut at the lower left corner between the two sectors, i.e. the cut C_t.*

Now consider $k_1 = k_2$. In Figure 11, p' is the ring number where $k_1 = k_2 = \cdots = k_{p'+1}$. We will show that a cut C is less dense than the cut C'. Since $k_{p'+1} = k_1$, $B_1' = (b-p')k_1 + (b-p'-\gamma_1)\,u(b-p'-\gamma_1) - (b-p'-\delta_1)\,u(b-p'-\delta_1)$. $B_2' = \sum_{i=1}^{p'} k_i = p'k_1$ according to Theorem 1. $B_3' = (h-p')k_1 + (h-p'-\gamma_{p'})\,u(h-p'-\gamma_{p'}) - (h-p'-\delta_{p'})\,u(h-p'-\delta_{p'})$ and $V' = V$.

Similar to the case $k_1 = k_2 + 1$,

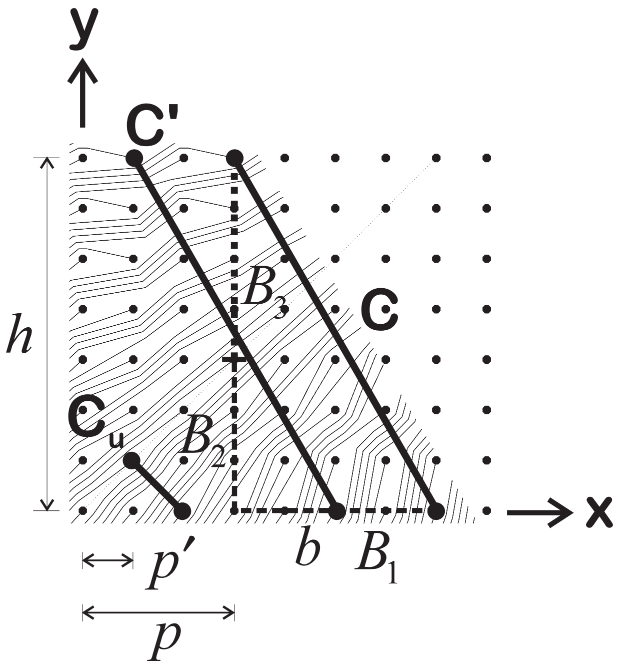

Figure 11: Extreme Position of C' when $k_1 = k_2$

$$\begin{aligned} B'_1 - B_1 &\geq -p \\ B'_3 + B'_2 - B_3 - B_2 &\geq (h-p')k_1 + p'k_1 - (h-p)(k_1 - 1 + 1) - p(k_1 - 1) = p. \end{aligned}$$

We have $F(C') - F(C) = B'_1 + B'_3 - V' - B_1 - B_2 - B_3 + V \geq 0$. Hence we have the following lemma.

Lemma 9: *If $k_1 = k_2$, the density of any cut is less than or equal to the density of the cut that ends on ring 1 and ring p' with the same slope.*

Since $k_{p'+1} = k_1$, we can maximize the density of C' with the same expression as we have done for the case $k_1 = k_2 + 1$. The densest cut is C_u (Figure 12) is in the grid cell of the base of C'. The flow of C_u is $2(k_{p'+1} - 1) = 2(k_1 - 1)$ which is the flow of C_s (Figure 9).

Combining the results for both cases, we have the following lemma.

Lemma 10: *The densest cut across two sectors is either the diagonal cut, C_t, at the corner of the design or C_s, the diagonal cut at the lower left corner of a sector, with density equal to $\sqrt{2}(k_1 - 1)$.*

5.3 Cuts Between Three Sectors

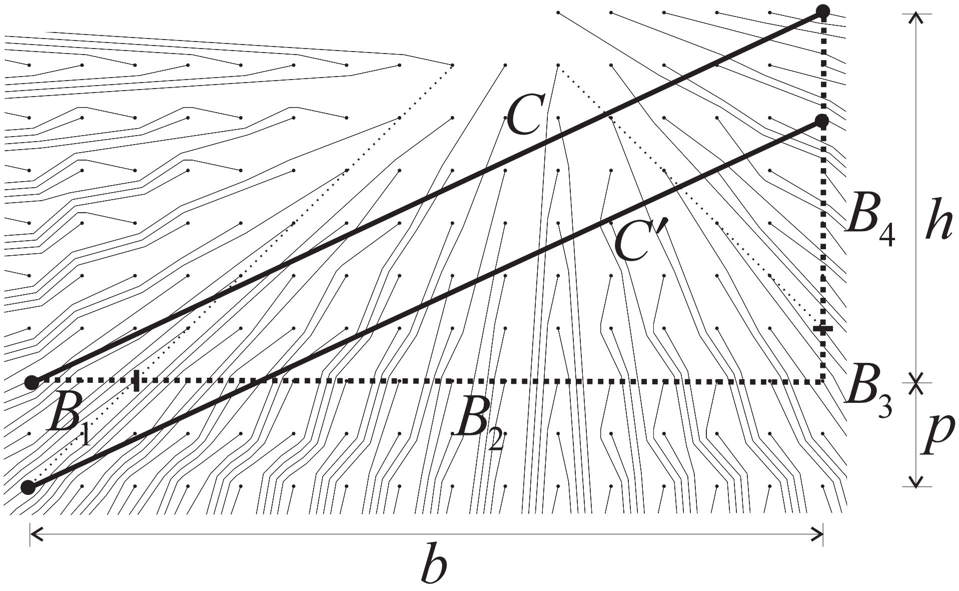

Figure 12: Cuts intersecting three sectors

We can use the same approach to consider cuts across three sectors. Figure 12 shows a cut C across three sectors. It is suffice to consider cuts with negative slopes and anchoring on ring 1 because of the symmetry of the package and the recursive nature of EVENPGA.

Consider the ring $p+1$. The cut C' which is two grids below C is more dense than C. We can verify this by looking at the cuts B_1, B_2, B_3 and B_4. The flow of B_1 increase because the side flows increase along the ring (Theorem 1). The increase of B_2 is at least $b-p$. This is because $k_r - k_{r+2} \geq 1$ by Lemma 2. B_3 is either unchanged or increased because the side flow of the new grid cells intersected by B_3 can be zero. B_4 decrease at most $k_{p+1}+1$. Since k_{p+1} is less than or equal to $R-p$, the increase in B_2 more than compensate the decrease in B_4. This is because $b \geq R$ so $b-p \geq R-p \geq k_{p+1}$.

We can repeat this argument until either the anchor or the head of the cut is at the boundary of a sector. Then we can apply the argument for cuts across two sectors in the previous section to calculate the critical cut.

A similar argument can be applied to cuts where $b-p$ is larger than $P_{p+1}/4$.

From the above arguments we established the set of critical cuts of a design and their densities. We conclude this section with the following theorem.

Theorem 2: *The set of critical cuts of a design is* $\{C_s, C_t\}$ *when* $k_1 \geq 4$. *The critical density is* $\sqrt{2}(k_1 - 1)$. *When* $k_1 < 4$, *the critical cut set is* $\{(\pi_i^1, \pi_{i+1}^1) | \gamma_1 < i \leq \delta_1\}$ *with density equal to* k_1.

6 Implementation and Results

Figure 13 and 14 shows a BGA package routed with EVENFANOUT. The algorithm is implemented as a router module in Surf. It only creates a topological routing. Design rule check is done by Surf automatically using the method described by Dai et al[4]. Surf also provides the user interface.

The time required to do the assignment is negligible compared to generating the rubber band topological routing.

Surf serves as the routing tool to route packages designed by PEART, the Package Early-Analysis and Routing Tool. Given a set of parameters such as total number of pins, number of rings and the pin and wire pitches, PEART creates the pin array and determines its routability by checking the appropriate cuts. The next phase will be to optimize a given parameter such as the number of rings under the constraint of routability.

7 Conclusion

In this paper we proposed an algorithm EVENFANOUT that assigns and routes the solder bumps of a Ball Grid Array package to a set of fanout points in a single layer. The algorithm takes $O(TS)$ time where T is the number of pins and S is the time required to generate one route.

Further, we found a set of necessary and sufficient critical cuts for any routing produced by EVENFANOUT. We also obtained the closed form expression of the bound of these cuts so that we can determine a package is routable before any routing is done.

8 Acknowledgment

The authors wish to thank David Staepelaere, Jeffrey Su and Tal Dayan for their excellent work on Surf on which the package router and PEART is built. We also like to thank Joel Darnauer for

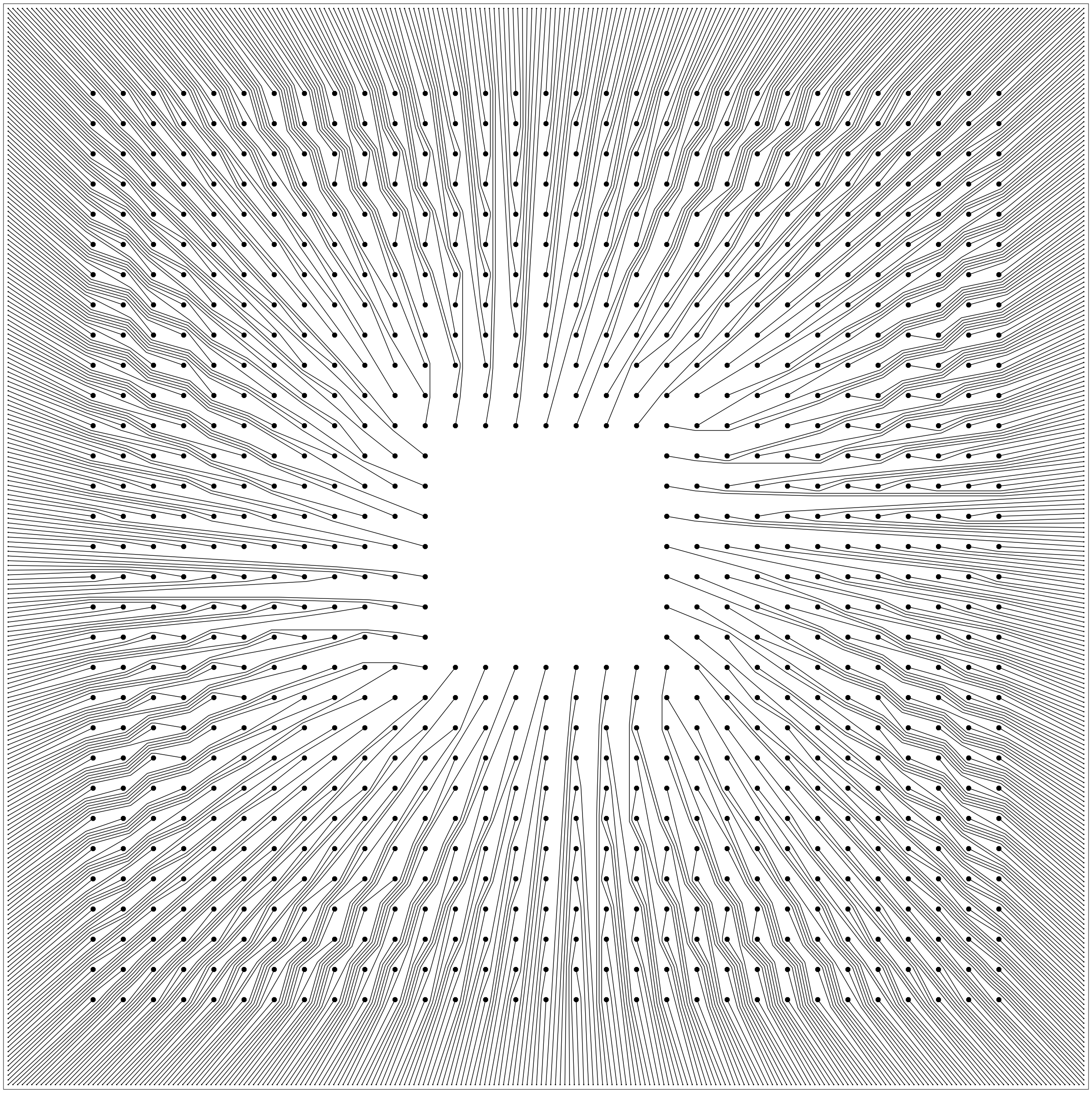

Figure 13: A 912 pin BGA with 12 rings and 31 pins per side

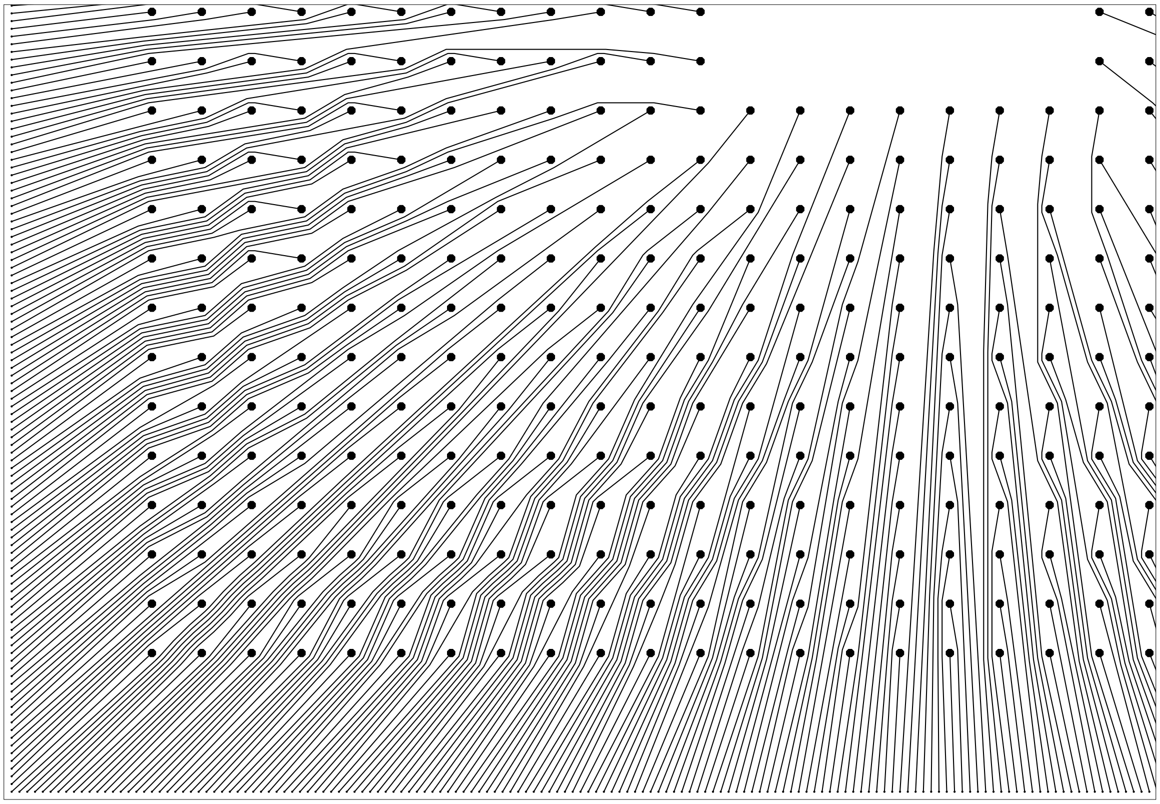

Figure 14: A quadrant of the 912 pin BGA

helpful discussions. We appreciate the help from AT&T Bell Labs for providing examples and help us understanding the problem.

References

[1] J. Darnauer and W. W.-M. Dai, "Fast pad redistribution from periphery-io to area-io," in *Proc. IEEE Multi-Chip Module Conf.*, (Santa Cruz, CA), pp. 38–43, March 1994.

[2] M. fai Yu and W. W.-M. Dai, "Pin assignment and routing on a single-layer pin grid array," Tech Report UCSC-CRL-95-15, University of California, Santa Cruz, 1995.

[3] W. W.-M. Dai, T. Dayan, and D. Staepelaere, "Topological routing in surf: Generating a rubber-band sketch," in *Proc. 28th Design Automation Conf.*, (Anaheim, CA), pp. 39–44, IEEE Computer Society Press, 1991.

[4] W. W.-M. Dai, R. Kong, and M. Sato, "Routability of a rubber-band sketch," in *Proc. 28th Design Automation Conf.*, (Anaheim, CA), pp. 45–48, IEEE Computer Society Press, 1991.

[5] F. M. Maley, *Single-layer wire routing and compaction.* Cambridge, MA: MIT Press, 1990.

[6] W. W.-M. Dai, R. Kong, J. Jue, and M. Sato, "Rubber band routing and dynamic data representation," in *Proc. 1990 Int'l Conf. on CAD*, (San Jose, CA), pp. 52–55, IEEE Computer Society, November 1990.

[7] M. fai Yu and W. W.-M. Dai, "Single-layer fanout routing and routability analysis for ball grid arrays," Tech Report UCSC-CRL-95-18, University of California, Santa Cruz, 1995.