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

Man-Fai Yu Wayne Wei-Ming Dai
Board of Studies in Computer Engineering
University of California, Santa Cruz, CA 95064

Abstract

Fanout routing for Ball Grid Array(BGA) packages becomes non-trivial when the I/O pin count increases. When the number of I/Os gets larger and larger, the number of I/Os we can put on a package may not 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 presents an efficient algorithm (EVENFANOUT) which generates the optimal uniform distribution of wires. We have found the three cuts that is decisive on the routability of the package using EVENFANOUT. These decisive cuts form the base for design optimization of the package.

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. Fig. 1 shows a BGA package with its fanout routing on the substrate.

A fully populated package may not be routable. We have to know how many pins we can put on a package before the fanout routing becomes unroutable. If we can determine the routability of a package without routing it, we can optimize the package for various goals such as to maximize the number of pins or minimize the size of the package.

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.

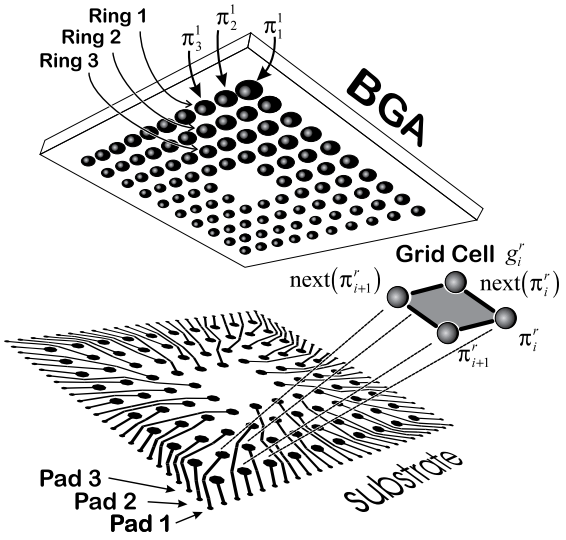

Figure 1: A Ball Grid Array Package and Its Fanout Routing on the Substrate

Our algorithm produces a routing at least as good as Darnauer and Dai[1]'s algorithm. The routing density is no more than $\sqrt{2}$ times the density of the perimeter cuts. Our algorithm is more easy to implement and takes advantage of the rubberband database of Surf.

2 Problem Definition

We define *pins* to be connectors on the package (BGAs, PGAs or area I/O chips) that are arranged in a grid array. *Pads* are via pads outside the pin array. 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 (Fig. 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 ring starts at the same corner as the pads. Since the pins are arranged in rings, all arithmetic involving the

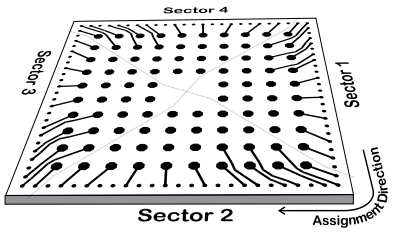

Figure 2: Pin Assignments in Each Sector of a Package

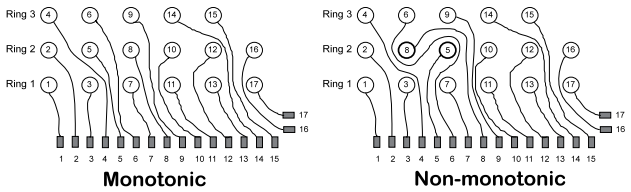

Figure 3: Monotonic vs. non-monotonic topological routing

subscript of a pin should be modulo P_r. To simplify the presentation, 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 (BGAFANOUT) can be defined as follows:

Problem 1 (BGAFANOUT) *The* single-layer fanout routing (BGAFANOUT) *problem is to create a single-layer 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.*

BGAFANOUT 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 BGAFANOUT. 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 wire* $w = (\pi_i^r, p)$ *connecting* π_i^r *and pad* p *intersect exactly one cut* (π_k^s, π_{k+1}^s) *for some* k *in each ring* $1 \leq s \leq r-1$.

This definition is similar to the MTR definition for PGA routing[2]. Fig. 3 compares an MTR and a non-MTR 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.

Algorithm 1 (EVENFANOUT)

Algorithm EVENFANOUT(Pins Π, Pads X)
 for rings $r \leftarrow 1$ to R {
  $k \leftarrow \lfloor |X|/P_r \rfloor$
  $q \leftarrow |X| - kP_r$, $t \leftarrow (P_r - q)/4$, $j \leftarrow 1$
  for pads $i \leftarrow 1$ to $|X|/4$ {
   ASSIGN4(i, j, r)
   $j \leftarrow j + 1$
   if $i \geq tk$ and $i < tk + q(k+1)$
    $i \leftarrow i + k + 1$
   else $i \leftarrow i + k$
  }
  Remove all assigned pads from X
 }

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

Figure 4: Algorithm EVENFANOUT

3 The EVENFANOUT Algorithm

The EVENFANOUT algorithm (Fig. 4) creates a monotonic pin assignment on a ring-by-ring basis starting from the outermost ring. It evenly distribute wires around the package to minimize congestion. The algorithm assigns pins in the clockwise direction in each sector (Fig. 2). 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. It is straightforward to verify that EVENFANOUT creates an MPA. After the assignment, we can generate the topological routing using the simple algorithm MAKEMTR (Fig. 5).

Algorithm 2 (MAKEMTR)
Algorithm MAKEMTR(Φ, X, Π)
 for $r \leftarrow 1$ to R
  for $i \leftarrow 1$ to P_r
   Route π_i^r to $\Phi(\pi_i^r)$

Figure 5: Algorithm MAKEMTR

The routing from π_i^r to $\Phi(\pi_i^r)$ is created by the shortest-path algorithm in Surf[3, 4, 5, 6]. The time complexity of MAKEMTR is $O(|X|S)$ where S is the complexity of creating rubberband 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.

Without loss of generality, we normalize the pin pitch to unit distance. Let $\text{next}(\pi_i^r)$ denote the pin on ring $r+1$ that is closest to π_i^r. We define a *grid cell* g_i^r to be the area bounded by the pins π_i^r, π_{i+1}^r, $\text{next}(\pi_i^r)$ and $\text{next}(\pi_{i+1}^r)$. 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 $(\text{next}(\pi_i^r), \text{next}(\pi_{i+1}^r))$, the *left side cut* to be $(\text{next}(\pi_{i+1}^r), \pi_{i+1}^r)$, 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. Fig. 1 shows a pin and its related grid cell.

The following lemma states 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. This lemma can be proved easily by induction on the ring r.

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. k is the basis for step size calculation in the algorithm. It choose to use either k or $k+1$ for step size. We can easily bound the difference of k and q in adjacent rings using Lemma 1.

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

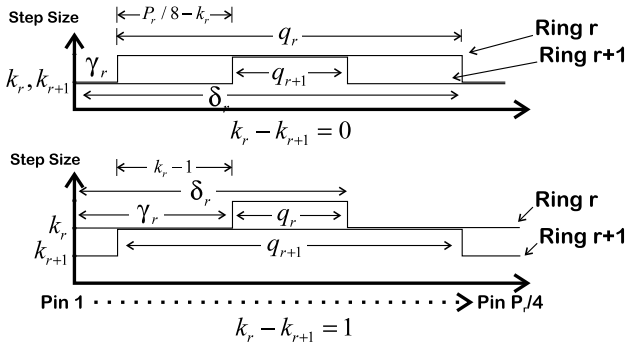

Figure 6: Different flows between adjacent rings

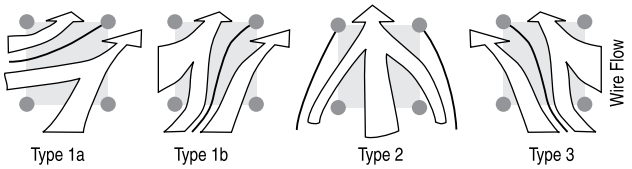

Figure 7: The flows inside a grid cell

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

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: Fig. 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.

Reg	j	T_i^r	B_i^r	$B_i^r - T_i^r$
I	$< \gamma_r$	k_{r+1}	k_r	0
II	$\gamma_r < j \leq \gamma_{r+1}$	k_{r+1}	$k_r + 1$	1
III	$\gamma_{r+1} < j \leq \delta_{r+1}$	$k_{r+1} + 1$	$k_r + 1$	0
IV	$\delta_{r+1} < j \leq \delta_r$	k_{r+1}	$k_r + 1$	1
V	$> \delta_r$	k_{r+1}	k_r	0

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 $L_i^r - R_i^r + a = B_i^r - T_i^r$. Fig. 7 shows the three types of cells based on how the pins in the cell is connected. The first cell (rightmost cell) in

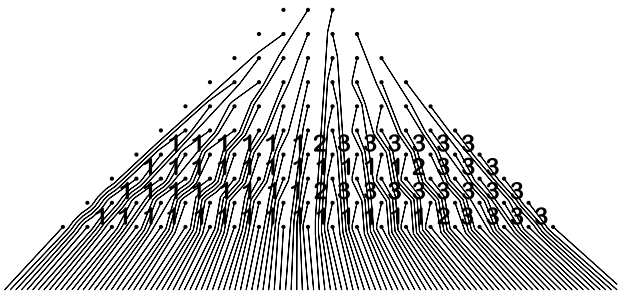

Figure 8: Cells in sector 2 and their types

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 (Fig. 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 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. Fig. 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 and the side flow changes monotonically with the difference of at most 1 along the ring.*

5 Routability Analysis

In this section we present a 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).

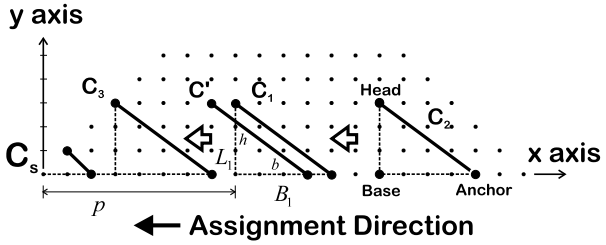

Figure 9: Cuts within a sector.

In the following we will adopt the grid coordinate for further analysis. We place the origin on the lower left corner of Sector 2 (Fig. 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.

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

Fig. 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.

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.

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. However, consider the grid cell of the base pin of C', g. The top and bottom flow is the same for this cell (Fig. 6). By Lemma 4, the difference of the left side cut and the right differs by 1. Therefore the flow across the left

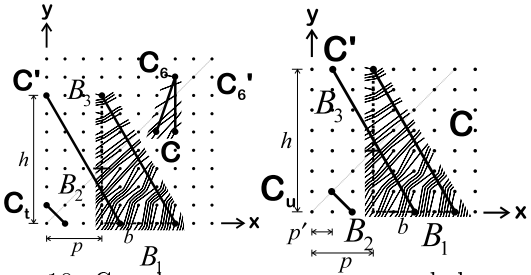

Figure 10: Cuts between two sectors and the extreme position of C' when $k_1 = k_2$

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 Fig. 9).

For a cut that touches the boundary, we can compute the upper bound of its density. The flow across the cut, $F = B + L - V \le b(k_1+1) + \sum_{i=1}^{h} k_i - \lfloor (b+1)(h+1)/2 \rfloor + 1$. 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. The density is therefore $D(C_3) = F/\sqrt{b^2+h^2}$ where b and h is the base and height of the triangle respectively. $D(C_3)$ maximizes at $b = h = 1$, i.e. the cut at the lower left corner of the sector, C_s. So we have

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

We can show that the densest cut is also C_s for the case $k_1 = k_2 + 1$. Combining the two 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 $k_1 \ge 4$, 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 $\gamma_1 < i \le \delta_1$. Their densities are k_1.*

The $k_1 \ge 4$ 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 $k_1 \ge 2 + \sqrt{2}$, the diagonal cut C_s dominates.

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 Fig. 10. We can immediately dismiss cuts with positive slopes like C_6 because $\Phi(\text{base}(C_6)) > \Phi(\text{anchor}(C_6)) > \Phi(\text{head}(C_6))$. The cut C'_6 is both shorter and captures more flow so it is denser than C_6. We only need to consider cuts with negative slopes. The flow across a cut C is $F(C) = B_1 + B_2 + B_3 - V$ (Fig. 10) where V is the number of connected pins inside the triangle.

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.

Cut	C	C'
B_1	$= bk_1 + \upsilon(b+p-\gamma_1) - \upsilon(b+p-\delta_1)$	$= bk_1 + (b-\gamma_1)\,\upsilon(b-\gamma_1) - (b-\delta_1)\,\upsilon(b-\delta_1)$
B_2	$\le \sum_{i=1}^{p} k_i$	$=0$
B_3	$= (h-p)k_{p+1} + \upsilon(h-p-\gamma_{p+1}) - \upsilon(h-p-\delta_{p+1})$	$= hk_1 + (h-\gamma_1)\,\upsilon(h-\gamma_1) - (h-\delta_1)\,\upsilon(h-\delta_1)$
V	$= \lfloor (b+1)(h+1)/2 \rfloor - 1$	$= \lfloor (b+1)(h+1)/2 \rfloor - 1$

In the above table, $\upsilon(x) = x\,\text{unitstep}(x)$.

Since $B'_1 - B_1 \ge -p$ and $B'_3 - B_2 - B_3 \ge hk_1 - p(k_1 - 1) - (h-p)(k_1 - 1 + 1) = p$, we have $F(C') - F(C) \ge 0$. $D(C')$ maximizes when $b = h = 1$ so we have

Lemma 7 *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 at the lower left corner of the package, C_t.*

Now consider $k_1 = k_2$. In Fig. 10, 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 $k_{p'+1} = k_1$, $B'_1 = (b-p')k_1 + \upsilon(b-p'-\gamma_1) - \upsilon(b-p'-\delta_1)$. $B'_2 = \sum_{i=1}^{p'} k_i = p'k_1$ according to Theorem 1. $B'_3 = (h-p')k_1 + \upsilon(h-p'-\gamma_{p'}) - \upsilon(h-p'-\delta_{p'})$ and $V' = V$.

Similar to the case $k_1 = k_2 + 1$, we have $F(C') - F(C) \ge 0$. Since $k_{p'+1} = k_1$, $D(C')$ maximizes to $D(C_u)$ which is equal to $D(C_s)$ (Fig. 9). Combining all the results, we have the following lemma.

Lemma 8 *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

We can use the same approach to consider cuts across three sectors. Fig. 11 shows a cut C across three sectors. It is suffice to consider cuts with negative slopes and anchoring on ring 1.

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

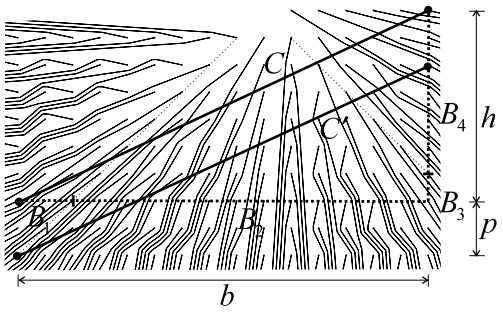

Figure 11: Cuts intersecting three sectors

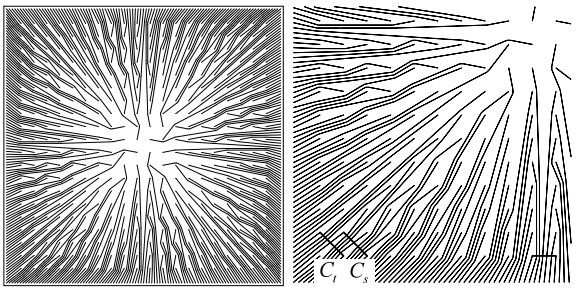

Figure 12: A 400 pin BGA with 10 rings and 20 pins per side and one of its quadrants

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 Results and Conclusion

Fig. 12 shows a BGA package routed with EVENFANOUT and the three critical cuts of Theorem 2. 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]. The time required to do the assignment is negligible compared to generating the rubber band topological routing.

In this paper we proposed an algorithm EVENFANOUT that assigns and routes the solder bumps of a BGA 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.

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.-F. 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] 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.

[6] D. Staepelaere, J. Jue, T. Dayan, and W. W.-M. Dai, "Surf:a rubber-band routing system for multichip modules," *IEEE Design and Test Magazine*, December 1993.