Net-aware Critical Area extraction for VLSI opens via Voronoi diagrams

Evanthia Papadopoulou*

Abstract

We address the problem of computing critical area for opens in a circuit layout in the presence of loops and redundant interconnects. The extraction of critical area is the main computational problem in VLSI yield prediction for random manufacturing defects. Our approach first models the problem as a graph problem and solves it efficiently by exploiting its geometric nature. The approach expands the Voronoi critical area computation paradigm [10, 7] with the ability to accurately compute critical area for missing material defects in a net-aware fashion. Generalized Voronoi diagrams used in the solution are combinatorial structures of independent interest.

1 Introduction

Catastrophic yield loss in integrated circuits is heavily attributed to random particle defects interfering with the manufacturing process resulting in functional failures such as open or short circuits. Random defect yield loss has been studied extensively resulting in several yield models (see e.g. [12, 11, 1]). The focus of all random defect yield models is the concept of critical area, a measure reflecting the sensitivity of a design to random defects during manufacturing. Reliable critical area extraction is essential for IC manufacturing especially when *design for manufacturability* (DFM) initiatives are under consideration.

The critical area of a circuit layout on a layer A is defined as

$$A_c = \int_0^{\infty} A(r)D(r)dr$$

where $A(r)$ denotes the area in which the center of a defect of radius r must fall in order to cause a circuit failure and $D(r)$ is the density function of the defect size. Critical area analysis is typically performed on a per layer basis and results are combined to estimate total yield. The defect density function has been estimated as follows [1, 5, 11, 14]:

$$D(r) = \begin{cases} cr^q/r_0^{q+1}, & 0 \le r \le r_0 \\ cr_0^{p-1}/r^p, & r_0 \le r \le \infty \end{cases} \quad (1)$$

where p, q are real numbers (typically $p = 3, q = 1$), $c = (q+1)(p-1)/(q+p)$, and r_0 is some minimum optically resolvable size. Using typical values for p, q, and c, the widely used defect size distribution is derived, $D(r) = r_0^2/r^3$. (r_0 is typically smaller than the minimum feature size thus, $D(r)$ is ignored for $r < r_0$). Following a common practice to facilitate critical area computation, a defect of size r is modeled throughout this paper as a square of radius r i.e., a square of side $2r$. Modeling defects as squares corresponds to computing critical area in the L_∞ metric. A formal bound for critical area between square and circular defects is given in [7]. For computational simplicity [9] the L_∞ metric is used throughout the paper.

In this paper we focus on critical area extraction for *opens* resulting from broken interconnects. The results are a generalization of the results presented in [7]. Opens are *net-aware*, that is, a defect forms a fault if it is actually *breaking* a net. A net is said to be broken if terminal points of the net get disconnected. An open circuit may also be caused by a *via-block* i.e., a defect on a via or contact layer that entirely destroys a connection (a via or cluster of vias) between neighboring conducting layers (see [7, 8]). In order to increase design reliability and reduce the potential for open circuits designers are increasingly introducing redundant interconnects creating loops that may span over a number of layers (see e.g. [4]). Redundant interconnects reduce the potential for opens at the expense of increasing the potential for shorts. The ability to perform trade-offs is important requiring accurate critical area computation for both. In this paper we accurately compute critical area for opens even in the presence of loops. Note that a loop reduces the potential for open faults but does not necessarily provide immunity to opens: a defect may still create an open by breaking two or more wires disconnecting the loop. The problem is modeled as a graph problem and solved efficiently by exploiting the geometric nature of it. A Voronoi diagram of the layout modeling opens allows accurate critical area computation. The algorithms are being integrated into the IBM Voronoi Critical Area Analysis tool (*Voronoi CAA*) currently used in production mode for chip manufacturing by IBM microelectronics (see [6] for results on the industrial use of the tool).

2 Problem Formulation

From a layout perspective a net is a collection of interconnected shapes spanning over a number of layers. Some of those shapes are designated as *terminal*

*IBM T.J. Watson Research Center, Yorktown Heights NY 10598, evanthia@acm.org

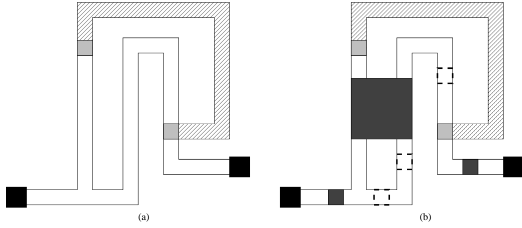

Figure 1: (a) A net N spanning over two layers. (b) Dark defects create opens while transparent defects are no faults.

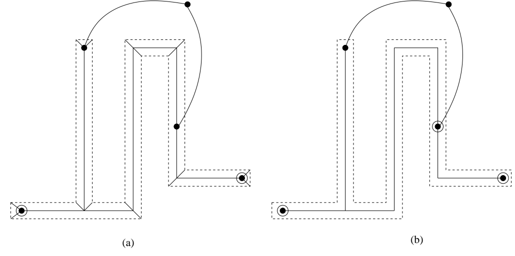

Figure 2: The net graph of Figure 1 before (a) and after (b) cleanup of trivial parts.

shapes representing the entities that must remain interconnected. A net remains *functional* as long as all terminal shapes comprising the net are interconnected. Otherwise the net is said to be *broken*. Figure 1(a) illustrates a simple net spanning over two metal layers (say M1 and M2, where M2 is illustrated shaded). The two contacts illustrated as black squares have been designated as terminal shapes. Figure 1(b) illustrates defects that create opens by breaking the net as dark squares and defects causing no fault as transparent squares in dashed lines.

Given a net N, the portion of N on a layer X, denoted $N_X = N \cap X$, consists of a number of connected components. Every connected component is a collection of overlapping shapes that can be unioned into a single polygon (a simple one or one with holes). A compact graph representation for N, denoted $G(N)$, can be defined as follows. There is a graph node for every connected component of N on a conducting layer such as Metal, Poly etc. A node containing terminal shapes is designated as a terminal node. Two graph nodes are connected by a graph edge iff there is at least one contact or via connecting the respective components of N. Given a layer A where critical area analysis needs to be performed, the *extended graph of* N on layer A denoted as $G(N, A)$ can be obtained from $G(N)$ by expanding all components of N_A by their medial axis. For every via or contact that connects a component of N_A to a component of N_B, $B \neq A$, introduce an approximate point along the medial axis representing that via or contact, referred to as a *via-point*. In addition, introduce an edge in $G(N, A)$ connecting the via-point with the respective node of N_B. If a contact or via has been designated as terminal shape, designate also the corresponding via point as terminal. Any portion of the medial axis induced by edges of terminal shapes is also identified as terminal. For the purposes of this paper we assume that $G(N)$ can be readily available by a built in net tracing capability. Then $G(N, A)$ can be easily obtained using well known Voronoi algorithms (e.g. [9] for L_∞). Net extraction is a well known topic beyond the scope of this paper.

We use the extended net graph $G(N, A)$ to detect loops. For this purpose we partition $G(N, A)$ into *biconnected components, bridges* and *articulation points* using *depth-first search* (DFS) as described in [13, 3]. For our problem we only need to keep some additional terminal information to determine whether the removal of a vertex or edge actually *breaks* $G(N, A)$ i.e., whether it disconnects $G(N, A)$ leaving terminals in both sides. Any bridges or articulation points that do not disconnect terminals are called *trivial* and can be removed form $G(N, A)$. Figure 2(b) illustrates the net graph after cleaning all trivial parts. Circles indicate terminal and articulation points.

Definition 1 *A defect D is a* minimal break *of a simple shape P if D breaks P into at least two pieces and D has minimal size, that is, if D is shrunk by $\epsilon \geq 0$ then D will be entirely contained in the interior of P. A piece of P may trivially consist of a single edge. A minimal break is called* strictly minimal *if it contains no other minimal break in its interior. A* break *is any defect totally overlapping a minimal break.[7].*

Definition 2 *A* minimal open *is a defect D that breaks a net N and D has minimal size, that is, if D is shrunk by $\epsilon > 0$ then D no longer breaks N. D breaks N if any two terminal shapes of N get disconnected or if a terminal shape itself gets destroyed. A minimal open is called* strictly minimal *if it contains no other minimal open in its interior. An* open *is any defect entirely covering a minimal open.*

Definition 3 *The center of an open D is a* generator point *for D weighted by the size (radius) of D. A generator point is of order $k, k \geq 1$, if D creates an open by breaking k polygonal paths (wires). A segment formed as a union of (kth order) generator points is called a (kth order)* generator segment.

Definition 4 *The* core *of the extended net graph $G(N, A)$ on layer A, denoted $core(N, A)$, is the set of all medial axis vertices, including articulation, via, and terminal points, and all medial axis edges except the standard $45°$ edges*[1]. *$G(N, A)$ is assumed to have*

[1]The $45°$ edges of the L_∞ medial axis bisecting axis parallel polygon edges are referred to as *standard* $45°$

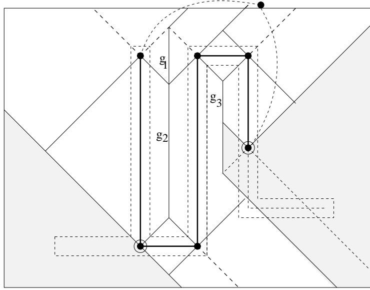

Figure 3: $V(C_i)$.

been cleaned up from any trivial components, trivial bridges, or trivial articulation points.

Lemma 1 *The 1st order generators for strictly minimal opens on layer A for net N, denoted $Gen_1(N, A)$, are the bridges, terminal edges, articulation points, and terminal points of $G(N, A) \cap core(N, A)$.*

2.1 Determining higher order generators for opens

Consider a bi-connected component of $G(N, A)$ denoted as C_i. Let $core(C_i) = C_i \cap core(N, A)$. For the purposes of this problem the endpoints and the open portion of a core segment are treated separately giving higher priority to the endpoints. Let the (weighted) Voronoi diagram of $core(C_i)$ be denoted $V(C_i)$, where any point p along a core segment s is weighted with $w(p) = d(p, e_l) = d(p, e_r)$, where e_l, e_r are the polygonal elements inducing s. Any Voronoi region equidistant from a core segment and its endpoint is assigned to the endpoint. Regions belonging to articulation or terminal points are colored red. See Figure 3.

We can identify the higher order generators for opens using higher order Voronoi diagrams of $core(C_i)$. In particular we can use $V^i(C_i), i \geq 1$, ($V^1(C_i) = V(C_i)$) to identify the $(i+1)$-order generators. It is not hard to see that a 2nd order strictly minimal open involving C_i must be generated by a point along a non-red, non-standard-45 Voronoi edge or a non-red Voronoi vertex of $V(C_i)$. The potential 2nd order generators of our example are illustrated in Figure 3 as $\{g_1, g_2, g_3\}$.

Let g be a Voronoi edge or vertex of $V(C_i)$ representing a potential opens generator. Let $core(g)$ be the set of core elements inducing g. There are three possible outcomes with respect to the connectivity of C_i after removing the elements of $core(g)$: 1) Removing $core(g)$ does not disconnect C_i; label *no-disconnect.* 2) Removing $core(g)$ disconnects the component but does not break it; label *trivial-disconnect.* 3) Removing $core(g)$ breaks the component; label *break.* If potential generator g is labeled break then g must be a 2nd order opens generator. Otherwise, g can not be a pure 2nd order generator (however portions of g might still be $k > 2$-order generators). The set of 2nd order generators determined form $V(C_i)$ by being labeled break is denoted $Gen_2(C_i)$. In figure 3 $Gen_2(C_i) = \{g_1, g_2\}$. All elements of $Gen_2(C_i)$ are colored red.

To determine ($k > 2$)-order generators (if any) we can repeat this process until all regions are colored red. In particular $V^k(C_i)$ can be obtained from $V^{k-1}(C_i)$ with an iterative process computing a slight modification of order k Voronoi diagrams. Red regions of $V^{k-1}(C_i)$ remain red in $V^k(C_i)$ and no further k-order subdivision is performed within. For every non-red region of $V^{k-1}(C_i)$ we compute the (slightly modified) k-order Voronoi subdivision as follows: Let H^{k-1} be the set of core elements owning the region of $V^{k-1}(C_i)$ under consideration ($reg(H^{k-1})$). Let $s(H^{k-1})$ be the superset of H^{k-1} including all core segments that are incident to core points in H^{k-1}. Let $N(s(H^{k-1}))$ denote the union of core elements owning Voronoi regions incident to regions of core segments in $s(H^{k-1})$ excluding any core elements already in $s(H^{k-1})$. Compute the (weighted) L_∞ Voronoi diagram of $N(s(H^{k-1}))$ and truncate it within the interior of $reg(H^{k-1})$. This gives the kth order subdivision within $reg(H^{k-1})$. Edges of $V^{k-1}(C_i)$ whose neighboring k-order regions have the same owners get removed from $V^k(C_i)$.

Given $V^k(C_i)$, $k \geq 1$, the set of potential $(k+1)$-order generators for strictly minimal opens must be the set of non-red, non-standard-45° Voronoi edges and vertices of $V^k(C_i)$. Thus, the set $Gen_{k+1}(C_i)$ of $(k+1)$-order generators for opens can be derived using a direct generalization of the process for $k = 1$.

To determine the labeling of Voronoi edges bounding a Voronoi region $reg(H)$ of $V^k(C_i)$, we can simply remove from C_i the tuple H of core elements owning $reg(H)$, and determine new bi-connected components, bridges and articulation points. In the case of 2nd order generators ($k = 2$) more advanced algorithmic techniques could be employed to derive a faster algorithm (see e.g. [2]). We do not attempt any such improvement in this simple version however. The time bound is $O(kn^2 + k^2 n \log n)$. In practice most biconnected components are sparse graphs due to the standard goal of minimizing wire length, in their majority just simple cycles, thus in any realistic situation k will be kept small. In case of biconnected components forming simple cycles the algorithm can easily simplify to $O(n \log n)$.

3 The Voronoi diagram for opens

Once generators for opens on a layer A are determined we can compute their Voronoi diagram and obtain a subdivision of the layout that allows fast critical area computation for opens.

Theorem 2 *Let $V(G)$ be the (weighted) L_∞ Voronoi*

diagram of the set G of all of generators for opens on layer A. The critical radius[2] *for opens for any point t on layer A is $r_c(t) = d_w(t, s)$, where $t \in reg(s)$ in $V(G)$.*

$V(G)$ is a generalization of the Voronoi diagram for breaks and via-blocks introduced in [7]. In L_∞ it is equivalent to the weighted Voronoi diagram of additively weighted segments, where the weight function along any segment s corresponds to an L_∞ distance from a line. For a Manhattan layout, generators are axis parallel segments and points with constant weights. The Voronoi region of a generator need not be connected, thus the size of $V(G)$ need not be linear.

Let K denote the number of *interacting* pairs of generators. A generator s_i is said to be *interacting* with generator s_j if $w(s_i) < w(s_j)$ and $R(s_i)$ intersects $R(s_j)$ in a non-trivial way, where $R(s)$ is the shape representing the union of all defects generated by a generator s. The intersection $R(s_i) \cap R(s_j)$ is non-trivial if it at least encloses an entire defect generated by some point $p \in s_i$. An upper bound for K is $O(n^2)$. In practice the number of interacting higher order generators must be small and thus K must also be small. A natural upper bound on the size of $V(G)$ is $O(n + K)$, where n is the complexity of layer A. $V(G)$ can be computed in $O((n + K) \log n)$ time by plane sweep by an algorithm similar to the one presented in [8] for the Hausdorff Voronoi diagram.

4 Critical Area Computation

Once the opens Voronoi diagram is available the entire critical area integral can be easily computed as shown in [10, 9, 7]. In particular, assuming the $1/r^3$ defect distribution, the critical area integral can be discretized as a summation of simple terms derived from Voronoi edges. For any other distribution, the Voronoi diagram still allows for analytical critical area integration within regions reducing the critical area integral to a summation of formulas. See [10, 9, 7] for details.

Critical Area for general missing material defects on layer A can be obtained by combining generators for opens on layer A and generators for via-blocks on the neighboring via/contact layers for the computation of the final opens Voronoi diagram. Generators for via blocks are axis parallel weighted segments (core segments) corresponding to portions of the farthest point Voronoi diagram of via clusters (see [7]). Thus, the Voronoi subdivision for opens on layer A can be easily extended to a Voronoi subdivision for general missing material defects combining opens and via blocks. Critical area computation can then be performed in an identical manner.

[2] The critical radius at a point t is the size of the smallest defect centered at t causing an open.

Acknowledgments

I would like to thank Dr. L. F. Heng of IBM T.J. Watson Research center for helpful discussions on the definition of a net graph. The IBM Voronoi EDA group including R. Allen, S.C. Braasch, J.D. Hibbeler and M.Y. Tan are greatly acknowledged for co-developing, supporting, and expanding the Voronoi Critical Area Analysis Tool.

References

[1] A. Ferris-Prabhu. Defect size variations and their effect on the critical area of VLSI devices. *IEEE J. of Solid State Circuits*, 20(4):878–880, Aug. 1985.

[2] J. Hopcroft and R. Tarjan. Dividing a graph into triconnected components. *SIAM Journal on Computing*, 1973.

[3] J. Hopcroft and R. Tarjan. Efficient algorithms for graph manipulation. *Comm. ACM*, 16(6):372–378, 1973.

[4] A. B. Kahng, B. Liu, and I. I. Mandoiu. Non-tree routing for reliability and yield improvement. *IEEE Trans. on Comp. Aided Design of Int. Circuits and Systems*, 23(1):148 – 156, 2004.

[5] I. Koren. *Yield Modeling and defect Tolerance in VLSI circuits*, "The effect of scaling on the yield of VLSI circuits", pages 91–99. Adam-Hilger Ltd., 1988.

[6] D. N. Maynard and J. D. Hibbeler. Measurement and reduction of critical area using Voronoi diagrams. In *Advanced Semiconductor Manufacturing IEEE Conference and Workshop*, 2005.

[7] E. Papadopoulou. Critical area computation for missing material defects in VLSI circuits. *IEEE Trans. on Computer-Aided Design*, 20(5):583–597, 2001.

[8] E. Papadopoulou. The Hausdorff Voronoi diagram of point clusters in the plane. *Algorithmica*, 40:63–82, 2004.

[9] E. Papadopoulou and D. T. Lee. The l_∞ Voronoi diagram of segments and VLSI applications. *International Journal of Computational Geometry and Applications*, 11(5):503–528, 2001.

[10] E. Papadopoulou and D. T. Lee. Critical area computation via Voronoi diagrams. *IEEE Transactions on Computer-Aided Design*, 1999.

[11] C. H. Stapper. Modeling of defects in integrated circuit photolithographic patterns. *IBM J. Res. Develop.*, 28(4):461–475, 1984.

[12] C. H. Stapper and R. J. Rosner. Integrated circuit yield management and yield analysis: Development and implementation. *IEEE Trans. Semiconductor Manufacturing*, 8(2):95–102, May 1995.

[13] R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1972.

[14] I. A. Wagner and I. Koren. An interactive VLSI CAD tool for yield estimation. *IEEE Trans. on Semiconductor Manufacturing*, 8(2):130–138, 1995.