Voronoi diagrams for VLSI manufacturing:
Robustness and Implementation

Evanthia Papadopoulou

IBM T.J. Watson Research Center
Talk Overview

● VLSI Layout — VLSI Data
● $L_{\infty}$: simpler Voronoi diagram for VLSI applications (small degree)
● Critical Area Computation Problem
  Important problem in VLSI Yield Prediction
● Critical area $\prec$ variations of $L_{\infty}$ VoDs
● Plane sweep construction of $L_{\infty}$ VoD
● Voronoi-based CAD tool for Critical Area Extraction: to be used by IBM manufacturing in ’03
VLSI Layout

- Layers of different materials
  - **Layer**: shapes realizing devices (e.g. transistors, interconnect)
    - Gate: PC (polysilicon) \( \cap \) RX (diffusion)
    - Interconnect: M1, M2, M3. Contacts: CA, V1
- **VLSI Layout**: Compact **hierarchical** form following logical (not physical) hierarchy
- **Library cell**: group of shapes (e.g. inverter). May appear in hundreds of places
- **Hierarchical Layout**: compact, cell repetition using transistions. Logical, not physical hierarchy
- **Flat Layout** printed in Mask: millions of shapes. Not available
VLSI Data

- Vast majority of segments are axis parallel
  45° orientations very common. Other orientations possible. Constant orientations can be assumed.
- Coordinates on integer grid (very large integers)
- Degenerate configurations are the rule
- No perturbation techniques to avoid degeneracies: destroy axis parallel property
- Data volume of flat data: order of millions (even billions)
VLSI Proximity Problems

- Width-spacing interactions essential. E.g. noise, yield, polishing
- **Design Rule Check**: Check width-dependent spacing rules
- **Lithography**: Printing of shapes depends on width/neighbor interactions
- **Yield Prediction**: Critical Area estimation
- **Medial axis/Voronoi diagram**: Address robustness issue. Adapt to VLSI hierarchy
  - Robustness issue: use $L_\infty$ metric
  - Construction not hierarchical: Based on plane sweep.
    Never keep VoD in memory: only active portion near scanline
Critical Area Problem

**VLSI Yield**: Percentage of manufactured chips that are working over all chips manufactured

Defects: Dust particles, Contaminants on materials

High Yield ⇒ Profit  
Low Yield ⇒ Loss

**Yield Prediction**: Control Cost of Manufacturing

\[ Y = \left( 1 + \frac{d A_c}{\alpha} \right)^{-\alpha} \]

\( d \) = ave. # defects per unit area,  \( \alpha \) = a clustering parameter,  \( A_c \) = the critical area of layout

**Critical Area**: measure reflecting sensitivity of layout to defects
Defect Types

Two types of defects:
- Extra material $\Rightarrow$ Shorts
- Missing material $\Rightarrow$ Open circuits

Missing Material defects: Breaks, Via Blocks
Defect of size $r = \text{circle/square of radius } r$
Critical Area

$$A_c = \int_{0}^{\infty} A(r) D(r) \, dr$$

$A(r)$: area of critical region for defect size $r$

**Critical region**: locus of points where if a defect of radius $r$ is centered causes a circuit failure

$$D(r) = \frac{r_0^2}{r^3}:$$ density function of the defect size
Critical Region ($A(r)$) for Shorts

Critical Area:

$$A_c = \int_0^\infty A(r)D(r)\,dr \quad \text{where} \quad D(r) = \frac{r_0^2}{r^3}$$
Critical Region \((A(r))\) for Opens

Critical Area:

\[ A_c = \int_0^\infty A(r)D(r) \, dr \quad \text{where} \quad D(r) = \frac{r_0^2}{r^3} \]
Critical Region \((A(r))\) for Via-Blocks

Critical Area:

\[
A_c = \int_0^{\infty} A(r) D(r)\,dr \quad \text{where} \quad D(r) = \frac{r_0^2}{r^3}
\]
Critical Area via Voronoi diagrams

**Shorts**  $A_c \ll 2$nd order $L_\infty$ Voronoi diagram of polys

**Opens**  $A_c \ll (weightd) L_\infty$ Vor dgrm of MA segmts

**Via Blocks**  $A_c \ll L_\infty$ min-Max (Hausdorff) Voronoi dgrm of polys

  Combinatorial structure of independent interest

**Bound:**

\[
A_c^2 \leq A_c^\infty \leq 2A_c^e
\]

$L_\infty$ metric $\equiv$ square defect model

---

E. Papadopoulou, IBM T. J. Watson Research Center  
DIMACS Workshop Dec. 4-6, 2002  
12
Why $L_\infty$?

Algorithmic degree

Test computations eval. multivariate polynomials of arithmetic degree $\leq d$. Test computations bit precision: $db + O(1)$

(input $b$-bit integers)

$L_2$ in-circle test (segments):
degree $\leq 40$

$L_\infty$ in-circle test (segments):
degree $\leq 5$

VLSI shapes: slopes are small constants $\triangleright$ degree 1

Liotta, Preparata, Tamassia, 96

Burnikel 96

(Papadopoulou & Lee, 99,01)

E. Papadopoulou, IBM T. J. Watson Research Center

DIMACS Workshop Dec. 4-6, 2002  13
\( L_\infty \) Quadrants

north, south quadrant: 
\( L_\infty \) dist = vertical dist

east west quadrant: 
\( L_\infty = \) horizontal distance

Non-orthogonal segments: 
Consider six 45° rays
$L_\infty$ Bisector

Degeneracy: Collinear axis parallel points
Bisectors: Straight-line segments

Bisector of a point and a line in $L_\infty$

Consists of $\leq 4$ parts, one for each quadrant of $p$

The unbounded portions are always $45^\circ$ rays
$L_\infty$ Voronoi Diagram

Ortho45 polygons: 8 orientations: 0, $\infty$, $\pm 1$, $\pm 3$, $\pm \frac{1}{3}$-slope
$L_\infty$ Voronoi diagram of segments

- Straight-line skeleton
- Maintains proximity information in $L_\infty$
  (good enough for practical applications)
- Voronoi vertices on Rational coordinates
  (assuming input on rational coordinates)
  Ortho-45 shapes: integer coordinates after multiplying by 12
- In-circle test, degree 5 ($L_2$: degree 40)
  VLSI shapes: slopes small constants $\rightarrow$ degree 1
- $O(n \log n)$-time plane sweep algorithm
  (based on wavefront paradigm of Dehne and Klein 97 and original plane sweep of Fortune 87)
  Algorithmic degree: 7

Papadopoulou & Lee 99,01
**Plane Sweep**

$L_{\infty}$ Plane Sweep: Sweep-line $l$ sweeping from left to right

Papadopoulou & Lee 99,01

based on Fortune 87, Dehne & Klein 97

$P_t$: polygons, portions of polygons to the left of $l$

Maintain: $Voronoï(P_t \cup l)$

**Wavefront**: boundary of Voronoi cell of $l$

**Spike Bisectors**: bisectors incident to wavefront

**Boundary**: treated as spike bisector
Data-Structures

- Planar Subdivision Data Structure (PSD): Half-edge data structure\(^{(\text{const. degree})}\)
- Wavefront: Red-Black tree
  Parameterize spike endpoints \((X(t), Y(t))\) as center of implied square. Key: \(Y(t)\). Ties resolved by PSD order
- Event list: Priority queue
Events

- **Site Events**: endpoints of segments  
  *Priority*: $x_i$

- **Spike Events**: intersections of neighboring spike bisectors  
  *Priority*: right side of implied square

Handling of Events:

- **PSD Operations**: \textit{CreateFace, SplitFace, SplitEdge, ContractEdge}
- **Assignment of Voronoi/Geometric Data**: slope, owner, spike equation  
  constant orientations \Rightarrow lookup tables
- **Wavefront searches by 45-deg rays**
Example Event Handling

- Search wavefront for $a$, $b$
- Fix PSD vertices between $a$, $b$ using $(X(t), Y(t))$
- Update PSD: create four new faces
- Assign Voronoi data: slopes, owners, new spike equations
Example of Site Event

- Update PSD
- Assign Voronoi data: new slope, spike equation
Voronoï Critical Area Tool

Layout

Scan Line
Online Shape
Overlap
Computation

Start/End
Edge Events

Voronoï Shorts
Voronoï Opens
Voronoï Via Blocks
Combination
Faults

Voronoï Region
Closings

Critical Area
Computation

Faults Probability

Legenda
I/O objects
computations
Some Experimental Results

Machine: RS6000/PPC bi-processor machine, processor speed 200MHz, memory 8 GBs (single-processor run)

Window: 1200 x 1200 microns about 13% chip

<table>
<thead>
<tr>
<th>Level</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time (min:sec)</td>
<td>3 : 18</td>
<td>0 : 41</td>
</tr>
<tr>
<td>Peak memory (MBs)</td>
<td>58.53</td>
<td>57.90</td>
</tr>
</tbody>
</table>
Some Experimental Results (continued)

**First Order Voronoi** (no face deletions)

<table>
<thead>
<tr>
<th>Time (min:sec)</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak memory (MBs)</td>
<td>7 : 55</td>
<td>1 : 11</td>
</tr>
<tr>
<td>N. Voronoi Faces</td>
<td>$3.0 \times 10^6$</td>
<td>$3.0 \times 10^5$</td>
</tr>
<tr>
<td>N. 2nd Order Voronoi Faces</td>
<td>$16 \times 10^6$</td>
<td>$1.0 \times 10^6$</td>
</tr>
</tbody>
</table>

**Shorts Voronoi**

<table>
<thead>
<tr>
<th>Time (min:sec)</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peak memory (MBs)</td>
<td>19 : 57</td>
<td>02 : 52</td>
</tr>
<tr>
<td>N. Voronoi Faces</td>
<td>$283.39$</td>
<td>$186.97$</td>
</tr>
<tr>
<td>N. 2nd Order Voronoi Faces</td>
<td>$4.4 \times 10^6$</td>
<td>$5.9 \times 10^5$</td>
</tr>
<tr>
<td>N. Polygons</td>
<td>$2.9 \times 10^5$</td>
<td>$6.3 \times 10^4$</td>
</tr>
<tr>
<td>Total N. Edges</td>
<td>$2.6 \times 10^7$</td>
<td>$3.2 \times 10^6$</td>
</tr>
</tbody>
</table>
Voronoi diagram for shorts

2nd order Voronoi diagram of polygons: every region has a *unique* owner which is responsible for shorts within the region

\[ A_c = \sum_V A_c(V), \quad V = \text{2nd order Voronoi cell} \]

(Papadopoulou & Lee 99,01)
Critical Area Integral

\[ A_c(\mathcal{R}) = \frac{r_0^2}{2} \left( \frac{l}{r_j} - \frac{l}{r_k} \right) \]

\[ A_c(T_{red}) = \frac{r_0^2}{2} \left( \ln \left( \frac{r_k}{r_j} \right) - \frac{l}{r_k} \right) \]

\[ A_c(T_{blue}) = \frac{r_0^2}{2} \left( \frac{l}{r_j} - \ln \left( \frac{r_k}{r_j} \right) \right) \]

\[ l = \text{length of vertical side, } r_k = \text{max critical radius, } r_j = \text{min critical radius} \]

Add up formulas ⇒ internal terms \( \frac{l_i}{r_i}, \ln \frac{r_k}{r_j} \) cancel out
Critical Area =
Summation of Voronoi edges

Critical area within $V$:

$$A_c(V) = \frac{r_0^2}{2} \left( \sum_{\text{red } e_i} \frac{l_i}{r_i} - \sum_{\text{blue } e_m} \frac{l_m}{r_m} + \sum_{\text{red } e_{45}} \ln \frac{r_k}{r_j} - \sum_{\text{blue } e_{45}} \ln \frac{r_k}{r_j} \right)$$

$\Rightarrow$ Exact Critical Area integration in one Layout pass $O(n \log n)$ Papadopoulou & Lee 99,01
Voronoi diagram for Opens

E. Papadopoulou, IBM T. J. Watson Research Center

DIMACS Workshop Dec. 4-6, 2002
Voronoi diagram for Via-Blocks

$L_\infty$ Min-Max VD of rectangles

Voronoi diagram under $d_{max}(t, R) = \max\{d(t, p), \forall p \in R\}$

Voronoi diagram of polygons under Hausdorff distance

E. Papadopoulou, IBM T. J. Watson Research Center

DIMACS Workshop Dec. 4-6, 2002 31
Conclusion

- Critical Area Extraction Tool
  - Based on $L_\infty$ Voronoi diagram of segments
  - Plane Sweep: unfolds layout hierarchy on fly
  - Data Volume of Chip: break into small windows

- Advantage of $L_\infty$ metric for VLSI applications

- Current/Future Work
  - Use statistics on many small windows
  - Take advantage of Layout repetition