Voronoi diagrams for VLSI manufacturing: Robustness and Implementation

Evanthia Papadopoulou

IBM T.J. Watson Research Center

Talk Overview

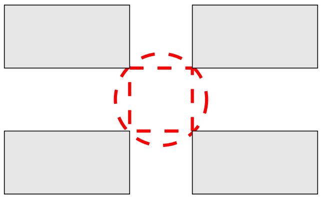
- VLSI Layout $-$ VLSI Data
- L_∞: simpler Voronoi diagram for VLSI applications (small degree)
- Critical Area Computation Problem
  Important problem in VLSI Yield Prediction
- Critical area $\prec$ variations of L_∞ VoDs
- Plane sweep construction of L_∞ VoD
- Voronoi-based CAD tool for Critical Area 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
- **ASIC chip**: thousands of interconnected cells (Many repeating cells)
- **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
  $45 \deg$ 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
- Our Voronoi tool for Critical Area
  - Robustness issue: use L_∞ 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 $\Rightarrow$ Profit
Low Yield $\Rightarrow$ Loss

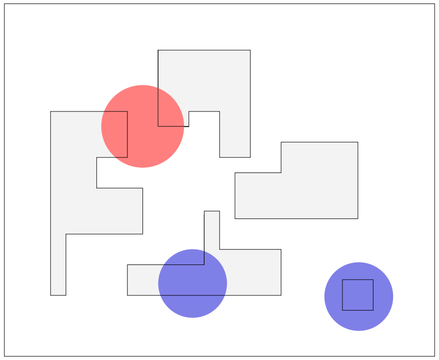
Yield Prediction: Control Cost of Manufacturing

$$Y = \left(1 + \frac{dA_c}{\alpha}\right)^{-\alpha}$$

$d =$ 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 = circle/square of radius r

Critical Area

Critical Area:

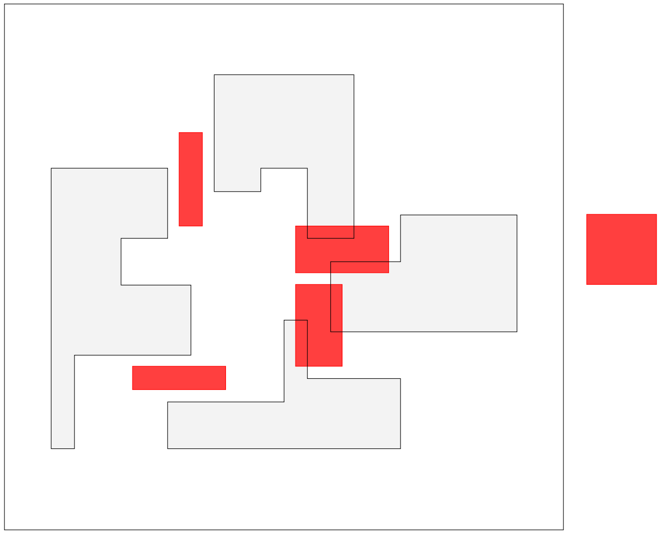
$$A_c = \int_0^\infty A(r)D(r)dr$$

$A(r)$: 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

$D(r) = r_0^2/r^3$: density function of the defect size

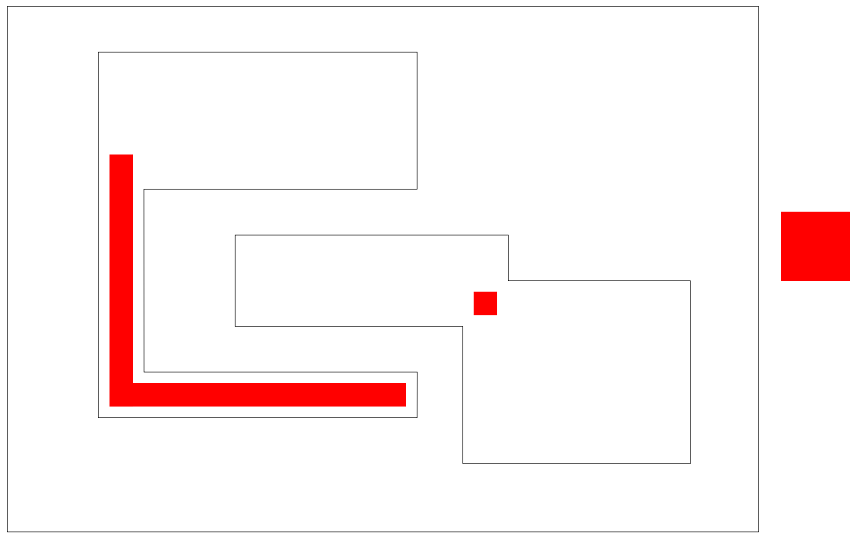
Critical Region ($A(r)$) for Shorts



Critical Area:

$$A_c = \int_0^\infty A(r)D(r)dr \quad \text{where} \quad D(r) = r_0^2/r^3$$

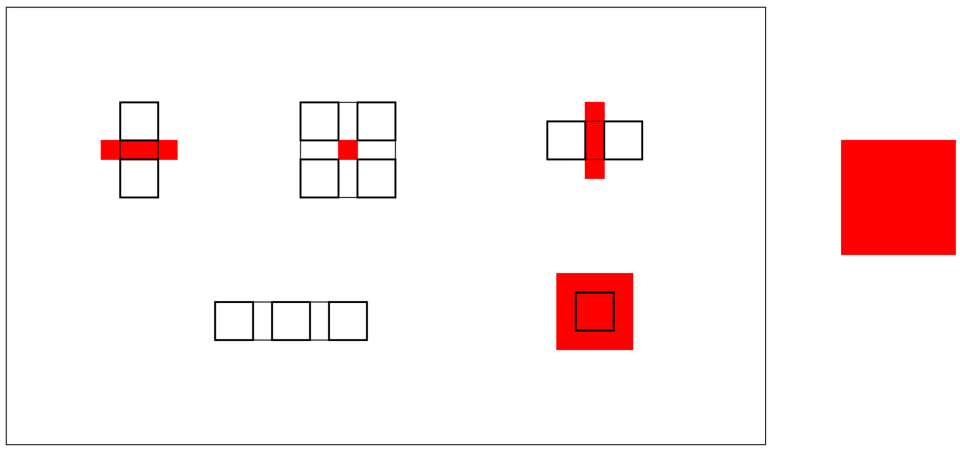
Critical Region ($A(r)$) for Opens



Critical Area:

$$A_c = \int_0^\infty A(r)D(r)dr \quad \text{where} \quad D(r) = r_0^2/r^3$$

Critical Region ($A(r)$) for Via-Blocks



Critical Area:

$$A_c = \int_0^\infty A(r)D(r)dr \quad \text{where} \quad D(r) = r_0^2/r^3$$

Critical Area via Voronoi diagrams

Shorts $A_c \prec$ 2nd order L_∞ Voronoi diagram of polys

Papadopoulou & Lee 99

Opens $A_c \prec$ (weightd) L_∞ Vor dgrm of MA segmts

Papadopoulou 01

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

Combinatorial structure of independent interest — Papadopoulou 01

$A_c = \sum($ terms derived from Voronoi edges$)$

Bound: $A_c^2 \leq A_c^\infty \leq 2A_c^e$

L_∞ metric $\equiv$ square defect model

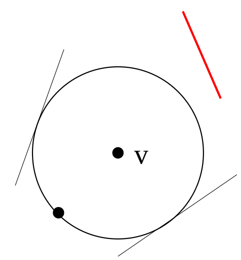
Why L_∞?

Algorithmic degree

Liotta,Preparata,Tamassia 96

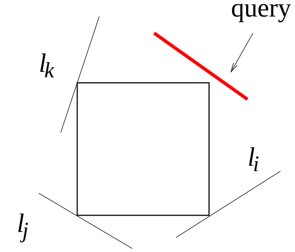
Test computations eval. multivariate polynomials of arithmetic degree $\leq d$. Test computations bit precision: $db + O(1)$ (input b-bit integers)

L_2 in-circle test (segments): degree ≤ 40



(Burnikel 96)

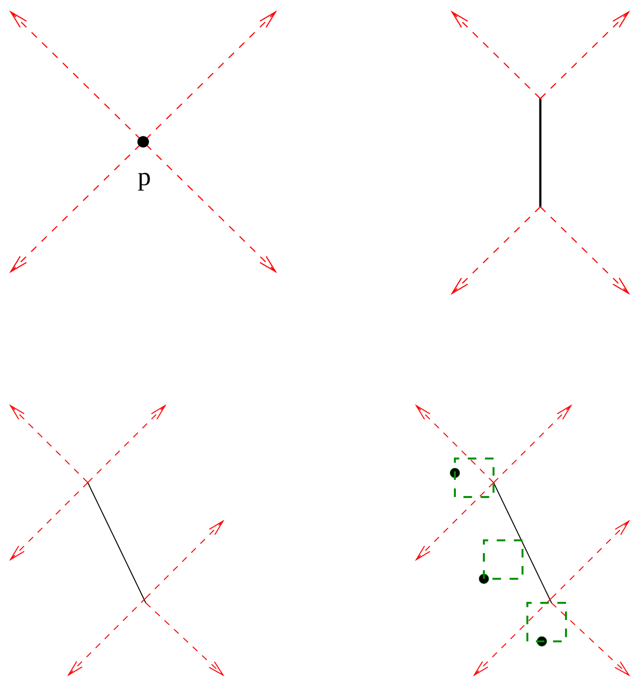
L_∞ in-circle test (segments): degree ≤ 5



(Papadopoulou & Lee 99,01)

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

L_∞ Quadrants

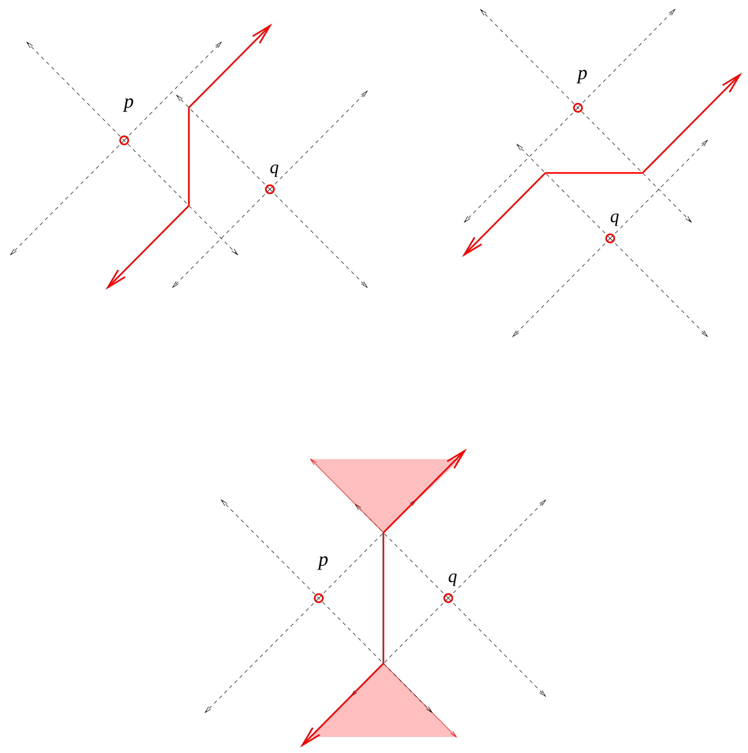


north, south quadrant:
L_∞ dist = vertical dist

east west quadrant:
$L_\infty =$ horizontal distance

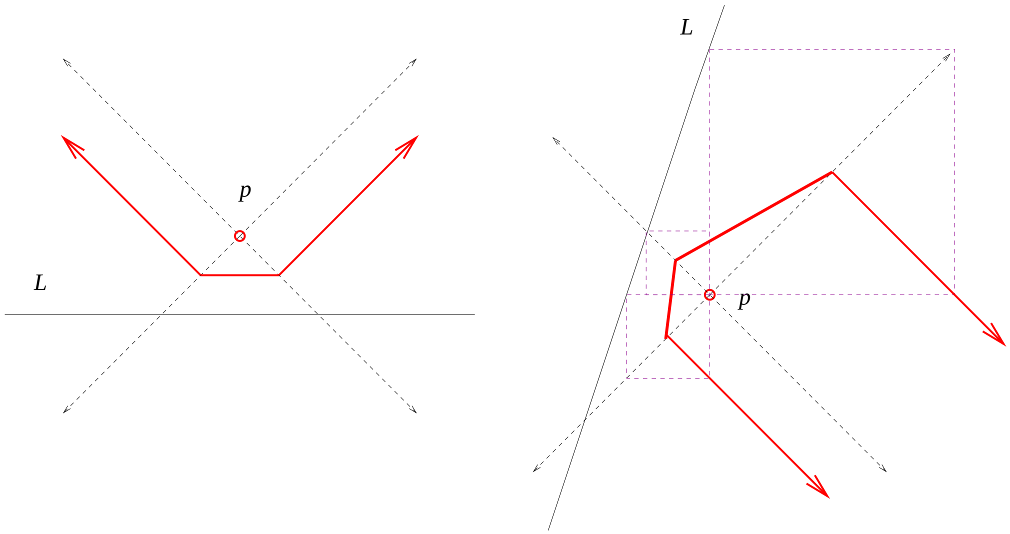
Non-orthogonal segments:
Consider six 45° rays

L_∞ Bisector



Degeneracy: Collinear axis parallel points

L_∞ Bisectors: Straight-line segments

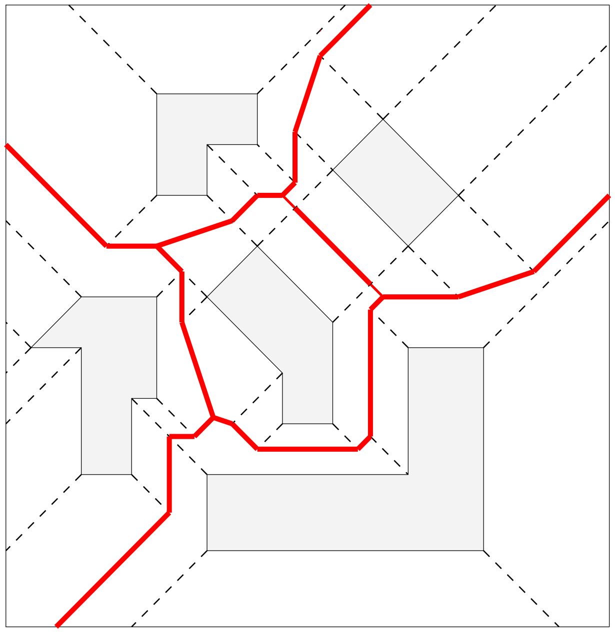


Bisector of a point and a line in L_∞

Consists of ≤ 4 parts, one for each quadrant of p

The unbounded portions are always 45° rays

L_∞ Voronoi Diagram



Straight-line skeleton

Ortho45 polygons: 8 orientations: $0, \infty, \pm 1, \pm 3, \pm \frac{1}{3}$-slope

L_∞ Voronoi diagram of segments

- Straight-line skeleton
- Maintains proximity information in L_∞

  (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)

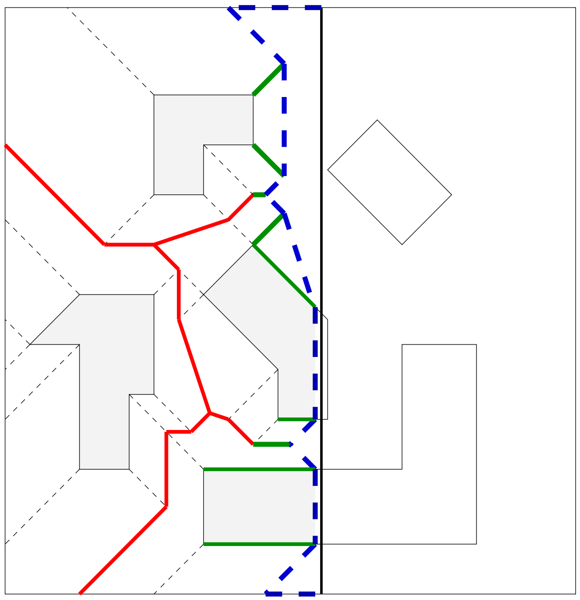
  VLSI shapes: slopes small constants $\rightarrow$ degree 1
- $O(n \log n)$-time **plane sweep** algorithm

  (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_∞ **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

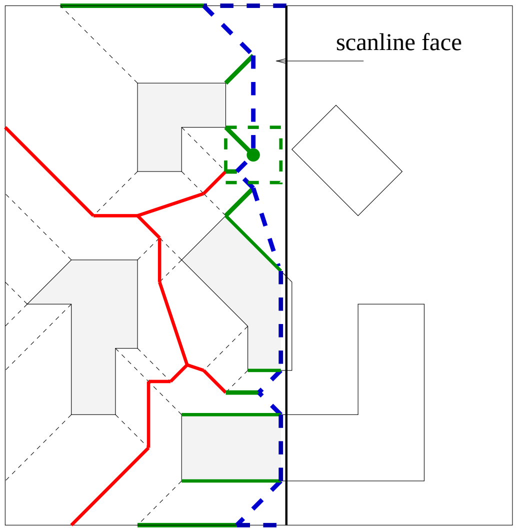
Maintain: *Voronoi*$(P_t \cup l)$

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 (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

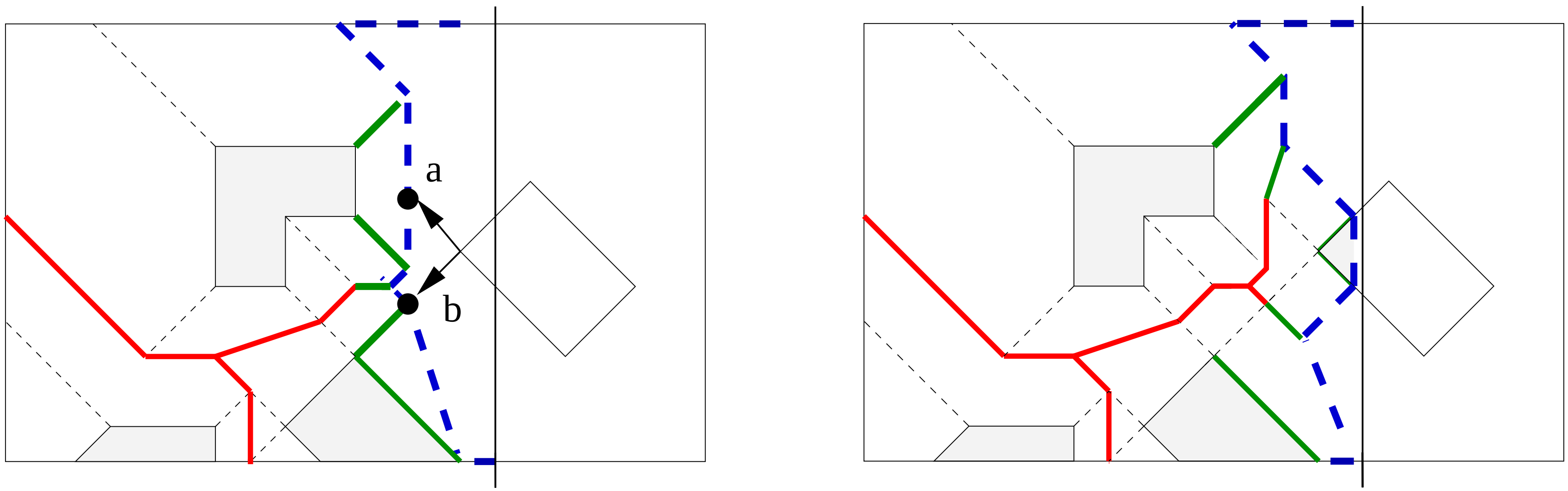
Events

- **Site Events**: endpoints of segments
  Priority: x_i
- **Spike Events**: intersections of neighboring spike bisectors
  Priority: right side of implied square

Handling of Events:

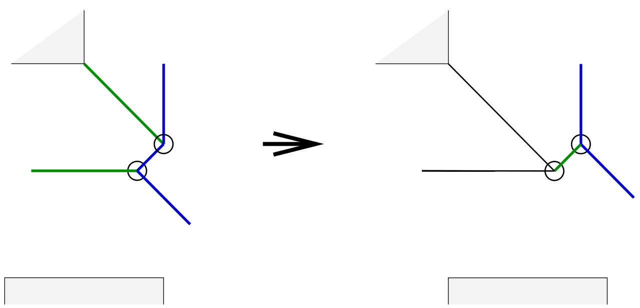
- PSD Operations: *CreateFace, SplitFace, SplitEdge, ContractEdge*
- Assignment of Voronoi/Geometric Data: slope, owner, spike equation
  constant orientations $\Rightarrow$ lookup tables
- Wavefront searches by 45-deg rays

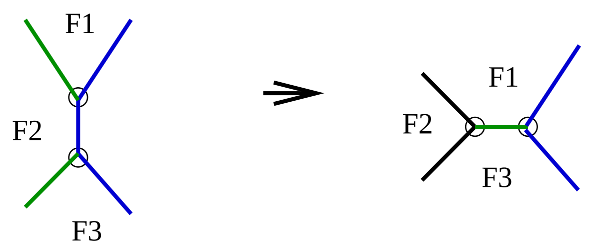
Example Event Handling



- 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

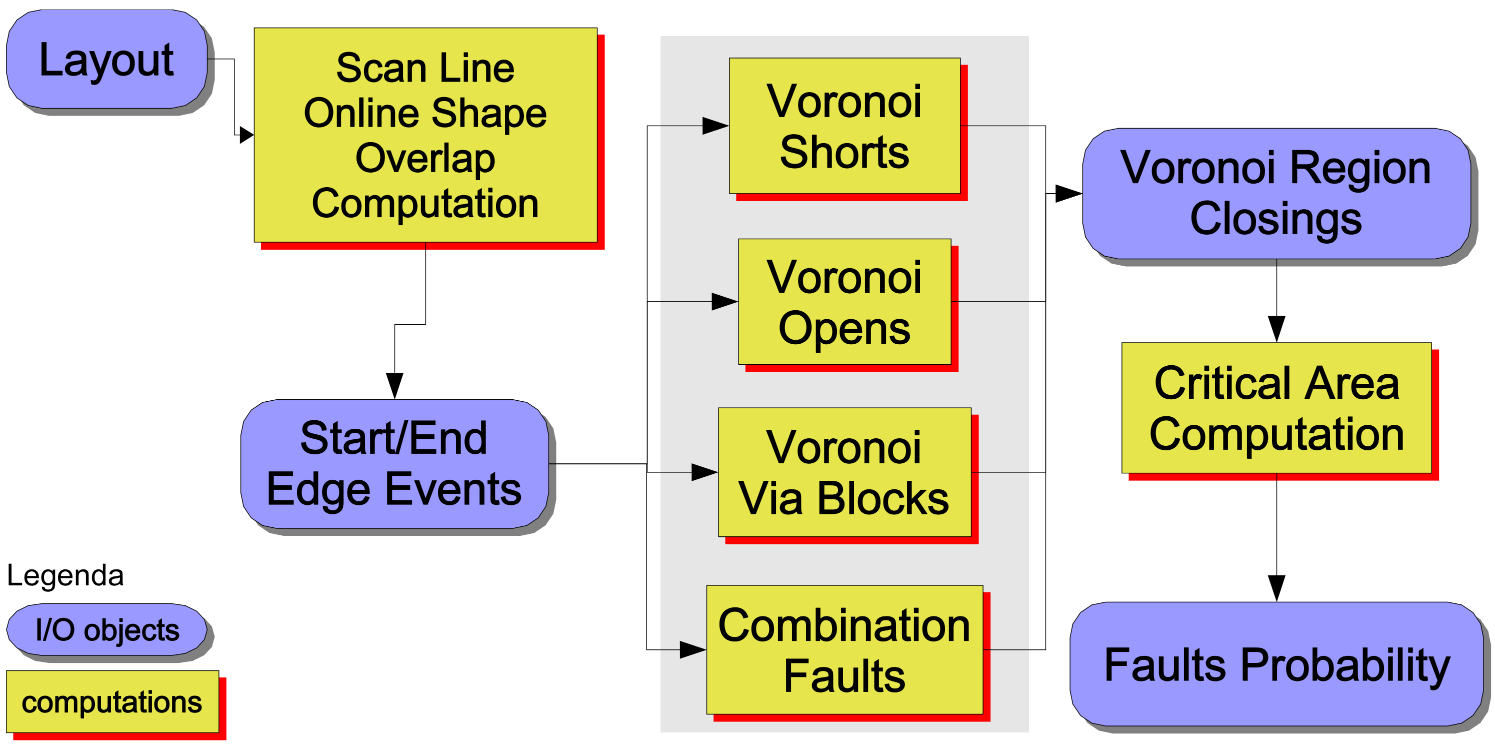
Example of Site Event





- Update PSD
- Assign Voronoi data: new slope, spike equation

Voronoi Critical Area Tool



Some Experimental Results

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

Window: 1200×1200 microns about 13% chip

Scan Line Processing

Level	M1	M2
Time (min:sec)	$3:18$	$0:41$
Peak memory (MBs)	58.53	57.90

Some Experimental Results (continued)

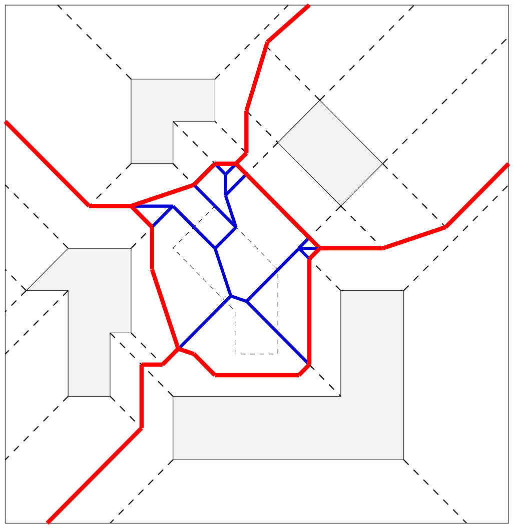
First Order Voronoi (no face deletions)

Level	M1	M2
Time (min:sec)	$7:55$	$1:11$
Peak memory (MBs)	904.13	197.72
N. Voronoi Faces	3.0×10^6	3.0×10^5
Total N. Edges	16×10^6	1.0×10^6

Shorts Voronoi

Level	M1	M2
Time (min:sec)	$19:57$	$02:52$
Peak memory (MBs)	283.39	186.97
N. 2nd Order Voronoi Faces	4.4×10^6	5.9×10^5
N. Polygons	2.9×10^5	6.3×10^4
Total N. Edges	2.6×10^7	3.2×10^6

Voronoi diagram for shorts



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

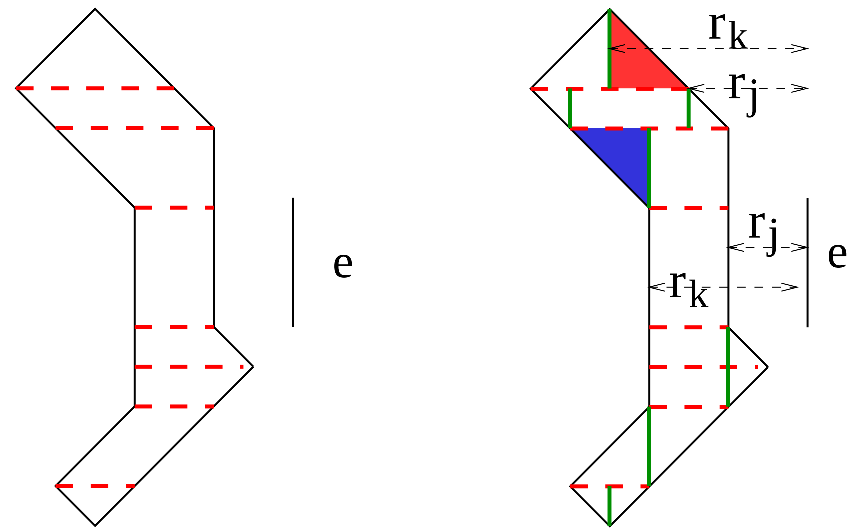
$A_c = \sum_V A_c(V)$, $V =$ 2nd order Voronoi cell (Papadopoulou & Lee 99,01)

Critical Area Integral

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

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

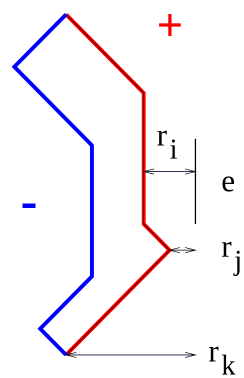
$$A_c(T_{blue}) = \frac{r_0^2}{2}\left(\frac{l}{r_j} - \ln\left(\frac{r_k}{r_j}\right)\right)$$



l = length of vertical side, r_k = max critical radius, r_j = min critical radius

Add up formulas $\Rightarrow$ internal terms $\frac{l_i}{r_i}$, $\ln\frac{r_k}{r_j}$ cancel out

Critical Area $=$
Summation of Voronoi edges

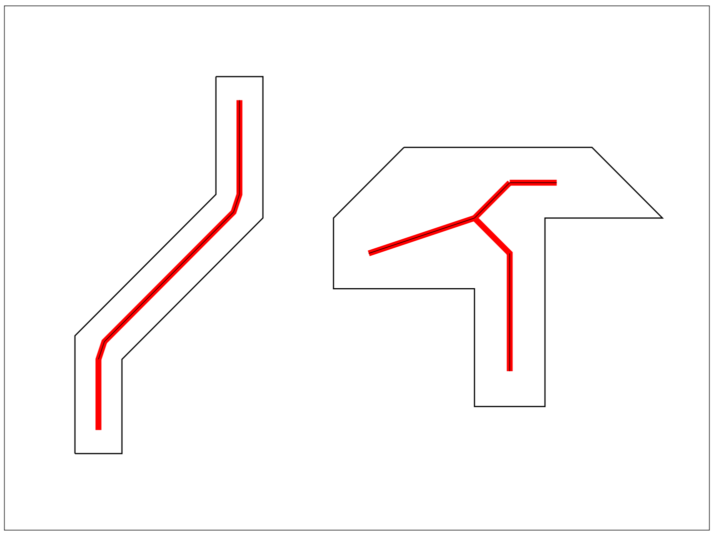


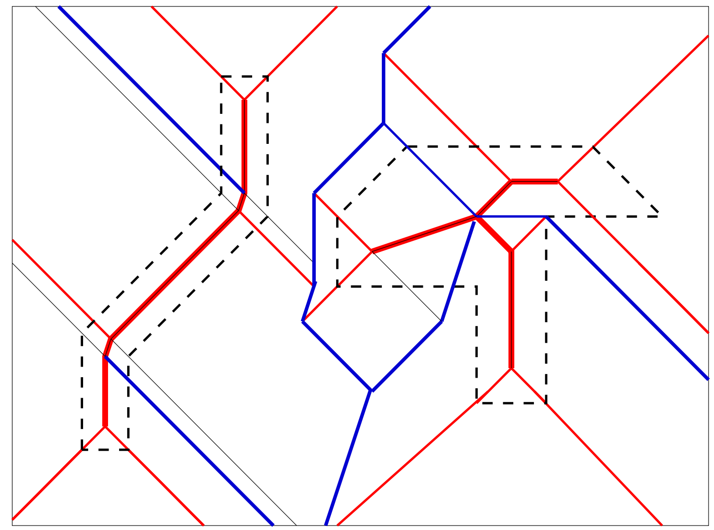
Critical area within V:

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

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

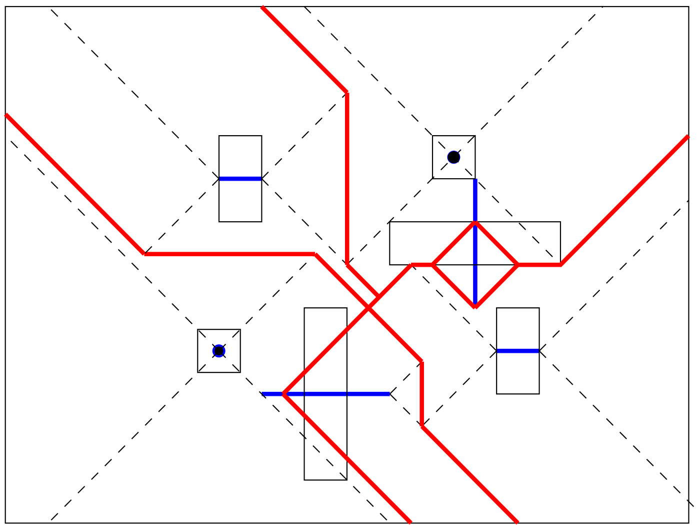
Voronoi diagram for Opens





Papadopoulou 01

Voronoi diagram for Via-Blocks



L_∞ Min-Max VD of rectangles — Papadopoulou 01

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

Voronoi diagram of polygons under Hausdorff distance — Papadopoulou & Lee 02

Conclusion

- Critical Area Extraction Tool
  - Based on L_∞ Voronoi diagram of segments
  - Plane Sweep: unfolds layout hierarchy on fly
  - Data Volume of Chip: break into small windows
- Advantage of L_∞ metric for VLSI applications
- Current/Future Work
  - Use statistics on many small windows
  - Take advantage of Layout repetition