Chip Finishing

Test Structures

- I-1 FORCE
- I-2 FORCE
- V-1 SENSE
- V-2 SENSE
- Metal
- Contact Hole
- Specified Wafer Layer
- 4-Point Probe Structure
- Van der Pauw Structure

Logos

Scribe

- Two Profile Break Wheel
- Wafer
- Film
- Mandrel
- Angle of Break Wheel exaggerated

Edge Seal
Making Chips

- Masks
- Wafers
- Chemicals
- Processing
- Processed Wafer
- Chips

Courtesy K. Yang, UCLA
Basic Fabrication: Two Steps

- (1) Transfer an image of the design to the wafer
- (2) Using that image as a guide, create the desired layer on silicon
  - diffusion (add impurities to the silicon)
  - oxide (create an insulating layer)
  - metal (create a wire layer)
- Use the same basic mechanism, photolithography, to do (1)
- Use three different methods to do (2)
  - Ion implant - used for diffusion: Shoot impurities at the silicon
  - Deposition - used for oxide/metal: Usually chemical vapor (CVD)
  - Grow - used for some oxides: Place silicon in oxidizing ambient
Photolithography

- **Repeat:**
  - Create a layer on the wafer (either before (oxide, metal) or after (diffusion) resist)
  - Put a photo-sensitive resist on top of the wafer
  - Optically project an image of the pattern you desire on the wafer
  - Develop the resist
  - Use the resist as a mask to prevent the etch (or other process) from reaching the layer under the resist, transferring the pattern to the layer
  - Remove the resist

- All die on the wafer are processed in parallel, and for some chemical steps, many wafers are processed in parallel
Photolithography

Start with wafer at current step

Spin on a photoresist

Pattern photoresist with mask

Step specific processing
etch, implant, etc...

Wash off resist

Courtesy K. Yang, UCLA
Photoresist Types

- Positive resists
  - material is removed from *exposed* areas during development
  - most widely used
- Negative resists
  - material is removed from *unexposed* areas during development
  - less mature

Post development profile for positive and negative photoresists
Mask Types

- **Bright field masks**
  - opaque features defined by chrome
  - background is transparent
  - used, e.g., for poly and metal

- **Dark field masks**
  - transparent features defined
  - background is opaque (chrome)
  - used, e.g., for contacts
  - used also for damascene metals
Mask (Reticle) Manufacturing

MEBES format and machine, or others

- Place a glass plate covered with chrome covered with resist in a high-vacuum column
- Use an electron beam spot size smaller than the finest resolution of the design
- Scan the surface of the mask with the e-beam in a raster-scan order. Modulate the beam to transfer the pattern to the chrome
- Develop the resist, and the chrome, and then remove the resist
- Check and correct the chrome pattern

All modern processes use masks (reticles) that are 5-10x larger than the desired size. The mask aligners then project the image and reduce it in the projection. While this means that exposing a wafer takes multiple prints, it is needed to reach the resolutions needed for current technologies.
OPC and PSM
Optical Proximity Correction (OPC)

- Layout modifications improve process control
  - improve yield (process latitude)
  - improve device performance

- Complicates mask manufacturing and increases cost

- Post-design verification is needed

![OPC Diagram](image-url)
Rule-Based OPC vs. Model-Based OPC

- **Rule-Based OPC**
  - Apply corrections based on a set of predetermined rules
  - Fast design time, lower mask complexity
  - Suitable for less aggressive designs

- **Model-Based OPC**
  - Use process simulation to determine corrections on-line
  - Longer design time, increased mask complexity
  - Suitable for aggressive designs
OPC Mechanisms

- Serifs: corner rounding
- Hammerheads: line-end shortening
- Gate assists (subresolution scattering bars): CD control
- Gate biasing: CD control
- Affects custom, hierarchical and reuse-based layout methodologies
OPC Issues

- WYSIWYG broken → (mask) verification bottleneck

- Pass functional intent down to OPC insertion
  - OPC insertion is for predictable circuit performance, function
  - Make only the corrections that win $$$ by reducing performance variation
    → cost-driven reticle enhancement technology (RET)

- Pass limits of manufacturing up to layout
  - don’t make corrections that can’t be manufactured or verified
  - Mask Error Enhancement Factor

- Layout needs models of OPC insertion process
  - geometry effects on cost of required OPC to yield function
  - costs of breaking hierarchy (beyond known verification, characterization costs)
Phase Shifting Masks

conventional mask

<table>
<thead>
<tr>
<th>Glass</th>
<th>Chrome</th>
<th>Phase shifter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 E at mask</td>
<td>0 E at wafer</td>
<td>0 I at wafer</td>
</tr>
</tbody>
</table>

phase shifting mask
Applicability of OPC and PSM
Gate Shrink
Forms of Bright-Field Alternating PSM

- **Single exposure**
  - phase transitions required, e.g., 0-60-120-180 or 90-0-270 to avoid printing phase edges
  - throughput unaffected
  - limited improvement in process latitude
  - mask manufacturing difficult, mask cost very high

- **Double exposure**
  - PSM with 0 and 180 degree phase shifters
  - define only critical features ("locally bright-field"), rest of mask is chrome
  - second exposure with clear-field binary mask protects critical features, defines non-critical features as well
  - better process latitude
  - decrease in throughput (double exposure)
PSM = Whose Problem?

- Must partition responsibility for phase-assignability into at least three domains

- Good layout practices
  - No T’s, no doglegs, even-length fingers on transistors, …
  - Open problem: What “design rules” guarantee phase-assignability without too much loss of density?

- Automatic phase conflict resolution

- Reuse of pre-existing layout
  - E.g., the entire standard-cell methodology is based on the assumption of “free composability” of cells within rows (as long as the cells don’t overlap)
  - Open problem: How to phase-assign layouts, such that no odd cycles of conflict occur when the layouts are composed
Phase Conflict and the Conflict Graph

- Self-consistent phase assignment is not possible if there is an odd cycle in the conflict graph.

- Phase-assignable = conflict graph is bipartite = no odd cycles
  - this is a global issue!
  - features on one side of chip can affect features on the other side.

- Breaking odd cycles: must change the layout!
  - change feature dimensions, and/or change spacings
  - Many degrees of freedom, e.g., layer reassignment for interconnects.
Conflict Graph

- **Dark Field:** build graph over feature regions
  - edge between two features whose separation is < B

- **Bright Field:** build graph over shifter regions
  - shifters for features whose width is < B
  - two edge types
    - **adjacency edge** between overlapping phase regions: endpoints must have same phase
    - **conflict edge** between shifters on opposite side of critical feature: endpoints must have opposite phase
Conflict Graph

- **Dark Field:**
  
  blue = feature; red = conflict

- **Bright Field:**
  
  conflict edge

  adjacency edge
### Conflict Graph for Cell-Based Layouts

- **Coarse view:** at level of connected components of conflict graphs within each cell master
  - each of these components is independently phase-assignable
  - can be treated as a single “vertex” in coarse-grain conflict graph
“Compaction-Based” PSM Layout Flow

- Analyze input layout
- Find min-cost set of perturbations needed to eliminate all “odd cycles”
- Induce shape, spacing constraints for new output layout
- “Compact” to get phase-assignable layout
- Goal: Minimize the set of new constraints, i.e., break all odd cycles in conflict graph by deleting a minimum number of edges
Conflict Edge Weight

- Which conflict edges are cheapest to break?
- Critical paths (e.g., in compactor) in x- and y-directions define layout area
- Conflict edges not on critical path: break for free
  - Criticality: with respect to, e.g., area or timing
Density Control for CMP
Flow Implications

- Accurate estimation of filling is needed in PD, PV tools (else broken performance analysis flow)
- Filling geometries affect capacitance extraction by > 50%
- Multilayer problem (coupling to critical nets, contacting restrictions, active layers, other interlayer dependencies)
Issues With Current Tools

- Only the average *overall* feature density is constrained, while local variation in feature density is ignored.

- Density analysis does not find *true* extremal window densities - instead, it finds extremal window densities only over fixed set of window positions.

- Fill insertion into layout does not minimize the maximum variation in window density.

- In part, due to physical verification tool heritage:
  - Boolean operations
  - Never empowered to change the layout.
Density Analysis
- find total feature area in each window
- find maximum/minimum total feature area over all $w \times w$ windows

- find slack (available area for filling) in each window

Fill synthesis
- compute amounts, locations of dummy fill
- generate fill geometries
Fixed $r$-Dissection Regime

- Feature area density bounds enforced only for fixed set of $w \times w$ windows
- Layout partitioned by $r^2$ distinct fixed dissections
- Each $w \times w$ window is partitioned in $r^2$ tiles
- How different is this from the regime of “continuous” window locations?

![Diagram](image-url)
Filling Problem
Filling Problem in Fixed-Dissection Regime

- **Given** design rule-correct layout of \( k \) disjoint rectilinear features in an \( n \times n \) layout region
- **Find** design rule-correct filled layout, such that
  - no fill geometry is added within distance \( B \) of any layout feature
  - no fill is added into any window that has density \( \geq U \)
  - minimum window density in the filled layout is maximized (or has density \( \geq \) lower bound \( L \))

- **Given**
  - fixed \( r \)-dissection of layout
  - feature \( \text{area}[T] \) in each tile \( T \)
  - \( \text{slack}[T] = \) area available for filling in \( T \)
  - maximum window density \( U \)

- **Find**
  - total fill area \( p[T] \) to add in each \( T \) s.t. any \( w \times w \) window \( W \) has density \( \leq U \)
  - \( \min_W \sum_{T \in W} (\text{area}[T] + p[T]) \) is maximized
Synthesis of Filling Patterns

- Given area of filling pattern $p[i,j]$, insert filling pattern into tile $T[i,j]$ *uniformly* over available area.

- Desirable properties of filling pattern:
  - uniform coupling to long conductors
  - either grounded or floating
Basket-Weave Fill Pattern

Each vertical/horizontal crossover line has same overlap capacitance to fill
Grounded Fill Pattern

Fill with horizontal stripes,
then span with vertical lines
Reticle Enhancement Roadmap

- Rule-based OPC
- Model-based OPC
- Scattering Bars
- AA-PSM
- Weak PSM
- Rule-based Tiling
- Optimization-driven MB Tiling

0.25 um | 0.18 um | 0.13 um | 0.10 um | 0.07 um

- Number Of Affected Layers Increases / Generation

248 nm
248/193 nm
193 nm

W. Grobman, Motorola – DAC-2001
Other Optical Lithography Issues

- Example: Field-dependent aberrations cause placement errors and distortions (→ location-specific cell variants?)

\[ \text{CELL}_A(X_1,Y_1) \neq \text{CELL}_A(X_0,Y_0) \neq \text{CELL}_A(X_2,Y_2) \]

Field-dependent aberrations affect the fidelity and placement of critical circuit features.
“$1M mask set” at 100nm, but average only 500 wafers per set
$1M NRE: Mask Write and Inspection Times

Context dependence: Same pattern, different fracture
$1M NRE: Mask Write and Inspection Times

- Too many data formats
  - Most tools have unique data format
  - Raster to variable shaped-beam conversion is inefficient
  - Real-time manufacturing tool switch, **multiple qualified tools → duplicate fractures** to avoid delays if tool switch required

- Data volume
  - OPC increases figure count acceleration
  - MEBES format is flat
  - ALTA machines (mask writers) slow down with > 1GB data
  - Data volume strains distributed manufacturing resources

- Refracturing mask data
  - 90% of mask data files manipulated or refractured: process bias sizing (iso-dense, loading effects, linearity, ...), mask write optimization, multiple tool formats, ...
ITRS Maximum Single Layer File Size
Mask Write Time vs. Data Volume

![Graph showing the relationship between ABF Data Volume (MB) and Write Time (Reformat + Print) (Hrs).](image)
Fracturing Problem

Mask Data Process Flow

Circuit Design → Tape Out → Mask Data Preparation → Mask Making

- Layout Extraction
- RET
- Fracturing
- Job Decomposition
- Tonality
- PEC Fracturing
- Job Finishing
- Writing
- Inspection
- Metrology
Challenges in Fracturing

• # shots increase
  mask writing time increase \(\rightarrow\) cost increase

• each shot should be an axis-parallel trapezoid
• the side size of each shot < \(M\)

• A shot whose minimum width < \(\varepsilon\) is called a sliver
• # slivers increase
  mask error enhancement factor (MEEF) increase
  larger CD variation and error \(\rightarrow\) yield decrease

• slant edges should not be partitioned
Fracturing Problem

Given:
- a list of polygons $P$ with axis parallel and slant edges
- Max shot size $M$
- Slivering size $\varepsilon$

Partition $P$ into non-overlapping trapezoidal shots

Minimizing:
Number of shots and number of slivers

Normal fracturing

Reverse tone fracturing
A “Ray Selection” Problem

• For each concave point (include inner point), choose one out of two candidate rays to minimize # slivers

Two candidates to kill one concave point

They are called as “conflict pair”
Multi-Project Wafers (= “Shuttle”)

- Amortize reticle, wafer costs by combining several designs from different projects onto one wafer (and, as few as three masks!)
- Share the expensive cost of a mask set
- Low-volume projects become feasible

Courtesy D. Bouldin, U. Tennessee
MPW Design Flow

Four Steps of Design Flow

1. Die Assignment
2. Placement
3. Shotmap
4. Stepper

Dicing Plan
Side-to-side Dicing Problem (SSDP)

Given:
- A die placement on the reticle
- A reticle placement on a wafer

Find:
- A set of horizontal and vertical cut lines

To Maximize:
- $z$, which is the minimal number of copies obtained across all the dies
H-Conflict

A die is obtained if and only if:
- Four edges are on the cut lines
- No cut lines pass the chip

Obtained: 1, 3

Two dies are in H-Conflict if no horizontal lines can cut off one without destroying the other

1 and 2 are in H conflict
Maximal Horizontal Independent Set

Maximal Horizontal Independent Set $S$ satisfy:
- Any two dies $\in S$ are not in H-conflict
- Any die $\not\in S$ is in H-conflict with at least one die $\in S$

Each Maximal horizontal independent set corresponds to a set of horizontal cutting lines, which are called Dicing Plan (DP).

MHIS: all maximal horizontal independent sets

$$MHIS=\{\{1,3\}, \{2,3\}, \{1,4\}, \{2,4\}\}$$
Reticle Design and Wafer Dicing Problem

Given: n dies $D_i$ (i=1…n),
Find: Placement of dies and a Dicing plan
To Minimize: Number of wafers used

20 wafers needed for 40 chips

40 wafers needed for 40 chips
Shot-Map and Stepping Problem

- Layout of reticle on multi-project wafer can have significant impact on die yield

- Objective: max weighted die yield
  - yield probability
    - defect distributions
    - edge loss
  - weights (target production volumes, etc.) for different designs
Yield Modeling - Defect Density Models

- **Poisson**
  \[ Y = e^{-AD} \]
  - random or even defect distribution

- **Murphy**
  \[ Y = \left( \frac{1 - e^{-AD}}{AD} \right)^2 \]
  - symmetrical triangular distribution \([0, 2D]\) with peak at \(D\)

- **Negative-Binomial (Seeds)**
  \[ Y = (1 + AD/\alpha)^{\alpha} \]
  - defect clustering
  - \(\alpha\): cluster parameter

- **Bose-Einstein**
  \[ Y = \frac{1}{(1 + AD)} \]
  - fatal defects: in certain critical mask layers
  - \(n\): # critical mask layers
  \[ Y = \left( \frac{1}{(1 + AD)} \right)^n \]
Edge radial yield degradation

- Wafer yields degrade with closer proximity of the wafer perimeter
- More important with larger wafer dimension
  - 200mm -> 300mm (average 12mmX12mm chip)
  - about 60% more chip bordering wafer perimeter

A unique radial yield model [Teets 94]

$$ Y_{radial} = \frac{1}{1 + e^{(\partial(r - \beta))}} $$

- $r$: distance of reticle from wafer center
- $\partial$ and $\beta$
  - empirically determined variables
  - functions of chip size
Reticle placement is optimized by taking into account many parameters that impact die yield:

- scribe lines,
- notches, flats
- wafer clamping
- weights for different designs
- yield probabilities
  - defect distributions
  - radial yield degradation