Faculty Presentation: Double-Patterning Impacts on Layout and Analysis

By Andrew B. Kahng, ECE and CSE Depts., UCSD

Mail: abk@ucsd.edu      Web: http://vlsicad.ucsd.edu/
Goals for Today

- Describe potential research directions related to double-patterning within the Design-Manufacturing Interface thrust of the IMPACT Center

- Focus on:
  - Automated layout decomposition for overlay robustness (with implications for layout practices, design rules, etc.)
  - Analysis implications of having two exposures

- Feedback, guidance from IMPACT colleagues

- Telephone (audio) backup if there is a problem:
  - Dial-in Number: 1-605-475-6300
  - Access Code: 966916
  - We will use this backup ONLY if the normal bridge becomes unusable
Double Patterning Lithography (DPL)

- ORAMEX (Ordinary Resist And Multiple Exposure), M. Neisser and A. Molless, IBM 1998
- Challenges: both equipment (e.g., overlay margin) and design (e.g., layout decomposition, DRC, new OPC for DPL)
Design-Side Issues of DPL (1)

- **Layout Decomposition**
  - Pattern features within the color spacing lower bound cannot be assigned different colors → one feature is divided into two parts
  - **Shortening and Rounding Awareness**
    - Decomposition results in more line-end shortening (LES) and corner rounding at edges of decomposed polygons
    - Guidelines for decomposition: (1) avoid creating jogging line-ends; (2) prefer cuts on landing pads and junctions; (3) ensure enough overlap at stitching points
  - **Overlay Awareness**
    - Objective function for polygon splitting should maximize overlay robustness

- **Hotspot detection and fixing**
- **Design compliance for DPL** (DPL-aware, grid-based, …)
- **Improvements to OPC** (topography, process type aware…)
- **Analysis** (“the bimodal problem”)
Layout is color-assignable if and only if
  – Whenever spacing of two features is less than the minimum color spacing (t), the features are assigned opposite colors

Example
  – Different colors: (n3, n5), (n2, n4) because d2,4, d3,5 < t
  – No need to assign different colors: (n1, n2), (n1, n3) because d1,2, d1,3 > t

Larger t $\rightarrow$ greater number of color conflicts
Conflict: Two features with < minimum color spacing have been assigned the same color

Like AltPSM phase conflict

- Could solve by spacing ("decompaction"), but layout iteration is undesirable
Layout Decomposition and Coloring

Layout Fracturing

Conflict Graph Construction

Conflict Cycle Detection

Conflict Cycle?

Yes

Node Splitting for Conflict Cycle Removal

No

Layout Partitioning

ILP based or heuristic Graph Coloring

Find min-cost color assignment
- Non-touching features with $0 < d(i,j) < t$ → different colors
- Touching features must have different colors → incur cost $c_{ij}$
Iterated Conflict Detection and Node Splitting

- Divided node may cause other conflict cycles → need iterative splitting until there is no conflict cycle

1st iteration of node splitting

2nd iteration of node splitting
Smart Splitting Point Selection

- Split polygons to have maximum overlap
- Reduce CD variation from LES, misalignment
Limits of Layout Decomposition

- Require DPL-compliant design
Examples of Unresolvable Conflict Cycles

- Example (a): No dividing point to remove the conflict cycle among all of rectangles (zero overlap length)
- Example (b): There is a dividing point, but overlap length < overlap margin
- DPL-compliant design: perturb layout to increase spacing
Min-Cost Color Assignment Problem

- **Given:**
  - A list of rectangles R which is color assignable, and maximum distance between two features, t, at which the color assignment is constrained

- **Find:**
  - A color assignment of rectangles to minimize the total cost

- **Subject to:**
  - For any two adjacent non-touching features with $0 < d(i,j) < t$, assign different colors
  - For any two touching features (i.e., $d(i; j) = 0$), if they are assigned different colors, there is a corresponding cost $c_{ij}$
  - $d(i,j)$: distance between features of $n_i$ and $n_j$
  - $x_i, x_j$: binary variables of features $n_i$ and $n_j$
  - $t$: minimum color spacing between features of $n_i$ and $n_j$
Color Assignment Example: Poly
Color Assignment Example: Metal
Layout Decomposition Summary

- **Color-assignment**
  - Successful color-assignment for design decomposition
  - Smart splitting point selection to maximize overlap length (e.g., 8nm at 45nm)
  - ILP color assignment algorithm minimizes #cuts and design rule violations
    - Layout partitioning to improve runtime

- **Ongoing works**
  - Variability-aware DPL decomposition
    - Hierarchical layout decomposition for DPL
    - Minimize difference between the pitch distributions of two masks
    - Minimize #distinct DPL solutions across all instances of a given master cell
  - Local, global density balance between two masks
Analysis Under Bimodal CD Distribution

- Bimodal CD distribution in DPL
  - Poly gates are made by two independent processes
  - Gate CD distributions in the two groups can differ

- Loss of correlations (spatial, line-space, launch-capture, clock-data, …) !!!

- **M\textsubscript{12} cell (NOR3X3)**
- **M\textsubscript{21} cell (NOR3X3)**


Figure 1. A simple bimodal distribution, in this case a mixture of two normal distributions with the same variance but different means. The figure shows the probability density function (p.d.f.), which is an average of the bell-shaped p.d.f.s of the two normal distributions.

Unimodal Modeling Is Too Pessimistic

- Conventional unimodal representation (cf. Dusa et al. 2007) does not capture bimodal process variation

- Key message: Our analysis and simulation studies (2008) suggest that DPL modeling, analysis, and yield optimization require close understanding of bimodal CD distribution
MC Gate Delay, Leakage: Simulation Setup

- **SPICE model**
  - 65nm, Typical corner (TT), 1.0V, 25°C

- **SPICE circuit**
  - 65nm NVT: Nominal CD is “60nm”

- **CD variation model**
  - Assumption: small mean difference
    - group1: $N(\text{mean}_{\text{M1}}=59\text{nm}, 3\sigma=5)$
    - group2: $N(\text{mean}_{\text{M2}}=61\text{nm}, 3\sigma=5)$

- **Comparison:**
  - \{Rise/Fall Delay, Leakage\} of unimodal and bimodal distribution

\[\begin{align*}
\text{Mean } G_1 &= 59 \text{ nm, } 3\sigma_{G_1} = 5 \text{ nm} \\
\text{Mean } G_2 &= 61 \text{ nm, } 3\sigma_{G_2} = 5 \text{ nm} \\
\text{Mean } \text{uni} &= 60 \text{ nm, } 3\sigma_{\text{uni}} = 6 \text{ nm}
\end{align*}\]
Simulation Results (MC iterations = 10,000)

- For bimodal distribution, two simulations are required.
  - DPL1: (2i-1)-th gate is group1 and 2i-th gate is group2
  - DPL2: 2i-th gate is group1 and (2i-1)-th gate is group2

- Characteristics of DPL1 and DPL2 are different; unimodal too pessimistic
Electrical Impact on Path Delay (1)

- Is bimodal distribution BAD or GOOD with respect to timing performance?

- Problem formulation:
  - Definitions
    - Let group1 and group2 be the sets of polys of the first and second patterning, respectively.
    - There are $m$ gates $g_i$ in group1 and $n$ gates $q_i$ in group2
  
  - Find delay variation of a given timing path, subject to
    - Covariance between gates in the same group is larger than the covariance between gates in different groups
      - $\text{Min}_{i,j} \text{cov}(g_i, g_j) > \text{Max}_{i,j} \text{cov}(g_i, q_j)$
      - $\text{Min}_{i,j} \text{cov}(q_i, q_j) > \text{Max}_{i,j} \text{cov}(g_i, q_j)$
By definition:

\[ \sigma^2(x+y) = \sigma^2(x) + \sigma^2(y) + 2 \text{ cov}(x,y) \]

Delay variation of a timing path:

\[
\sigma^2(d(\text{path})) = \sigma^2\left( \sum_i d(g_i) + \sum_j d(q_i) \right)
\]

\[
= \sum_i \sigma^2(d(g_i)) + \sum_i \sigma^2(d(q_i)) + 2 \sum_{i,j} \text{ cov}(d(g_i),d(g_j)) + 2 \sum_{i,j} \text{ cov}(d(q_i),d(q_j)) + 2 \sum_{i,j} \text{ cov}(d(g_i),d(q_j))
\]

Since \( \text{ cov}(d(g_i),d(q_j)) \) is smaller than \( \text{ cov}(d(g_i),d(g_j)) \) or \( \text{ cov}(d(q_i),d(q_j)) \), variation of bimodal distribution is smaller than unimodal distribution.

Path delay variation will be reduced

However, reduced variation does not necessarily imply good behavior from the design and signoff perspective.
Simulation Setup for Corner Cases (1)

- **Test cases**
  - Exhaustive simulation on both min and max corners of each CD group requires exponential number of simulations to the number of stages.

- Simple 4-stage delay chain: 2 (corners for CD group 1) x 2 (corners for CD group 2) x 2^4 (combinations of CD groups 1, 2 for 4 gates) = 64 cases

- **SPICE model**
  - 45nm PTM, Typical corner (TT), 1.0V, 25 °C

- **SPICE circuit**
  - 45nm INVX4 and BUFX4 from Nangate Open Cell Library
Simulation Setup for Corner Cases (2)

- **CD variation model**
  - Realistic bimodal with mean difference
  - Pooled unimodal:
    - CD model can account for physical CD distribution of entire population
    - CD variation increases with increasing mean difference.

- **Path sequences**

<table>
<thead>
<tr>
<th>Case</th>
<th>Inverter chain</th>
<th>Buffer chain</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case1</td>
<td>M1_M1_M1_M1</td>
<td>M12_M12_M12_M12</td>
</tr>
<tr>
<td>Case2</td>
<td>M1_M1_M1_M2</td>
<td>M12_M12_M12_M21</td>
</tr>
<tr>
<td>Case3</td>
<td>M1_M1_M2_M1</td>
<td>M12_M12_M21_M12</td>
</tr>
<tr>
<td>Case4</td>
<td>M1_M1_M2_M2</td>
<td>M12_M12_M21_M21</td>
</tr>
<tr>
<td>Case5</td>
<td>M1_M2_M1_M1</td>
<td>M12_M21_M12_M12</td>
</tr>
<tr>
<td>Case6</td>
<td>M1_M2_M1_M2</td>
<td>M12_M21_M12_M21</td>
</tr>
<tr>
<td>Case7</td>
<td>M1_M2_M2_M1</td>
<td>M12_M21_M21_M12</td>
</tr>
<tr>
<td>Case8</td>
<td>M1_M2_M2_M2</td>
<td>M12_M21_M21_M21</td>
</tr>
<tr>
<td>Case9</td>
<td>M2_M1_M1_M1</td>
<td>M21_M12_M12_M12</td>
</tr>
<tr>
<td>Case10</td>
<td>M2_M1_M1_M2</td>
<td>M21_M12_M12_M21</td>
</tr>
<tr>
<td>Case11</td>
<td>M2_M1_M2_M1</td>
<td>M21_M12_M21_M12</td>
</tr>
<tr>
<td>Case12</td>
<td>M2_M1_M2_M2</td>
<td>M21_M12_M21_M21</td>
</tr>
<tr>
<td>Case13</td>
<td>M2_M2_M1_M1</td>
<td>M21_M21_M12_M12</td>
</tr>
<tr>
<td>Case14</td>
<td>M2_M2_M1_M2</td>
<td>M21_M21_M12_M21</td>
</tr>
<tr>
<td>Case15</td>
<td>M2_M2_M2_M1</td>
<td>M21_M21_M21_M12</td>
</tr>
<tr>
<td>Case16</td>
<td>M2_M2_M2_M2</td>
<td>M21_M21_M21_M21</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Mean Diff</th>
<th>Uni-modal</th>
<th>M1</th>
<th>M2</th>
<th>M1</th>
<th>M2</th>
</tr>
</thead>
<tbody>
<tr>
<td>0nm</td>
<td>Pooled</td>
<td>50.00</td>
<td>2.00</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Bimodal</td>
<td>50.00</td>
<td>2.00</td>
<td>50.00</td>
<td>2.00</td>
</tr>
<tr>
<td>1nm</td>
<td>Pooled</td>
<td>50.00</td>
<td>2.50</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Bimodal</td>
<td>49.50</td>
<td>2.00</td>
<td>50.50</td>
<td>2.00</td>
</tr>
<tr>
<td>2nm</td>
<td>Pooled</td>
<td>50.00</td>
<td>3.61</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Bimodal</td>
<td>49.00</td>
<td>2.00</td>
<td>51.00</td>
<td>2.00</td>
</tr>
<tr>
<td>3nm</td>
<td>Pooled</td>
<td>50.00</td>
<td>4.92</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Bimodal</td>
<td>48.50</td>
<td>2.00</td>
<td>51.50</td>
<td>2.00</td>
</tr>
<tr>
<td>4nm</td>
<td>Pooled</td>
<td>50.00</td>
<td>6.32</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Bimodal</td>
<td>48.00</td>
<td>2.00</td>
<td>52.00</td>
<td>2.00</td>
</tr>
<tr>
<td>5nm</td>
<td>Pooled</td>
<td>50.00</td>
<td>7.76</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Bimodal</td>
<td>47.50</td>
<td>2.00</td>
<td>52.50</td>
<td>2.00</td>
</tr>
<tr>
<td>6nm</td>
<td>Pooled</td>
<td>50.00</td>
<td>9.22</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Bimodal</td>
<td>47.00</td>
<td>2.00</td>
<td>53.00</td>
<td>2.00</td>
</tr>
</tbody>
</table>
Path Delay of DPL (1)

- **Delay distribution across process corners (4-chains)**

  ![Graph showing delay distribution across process corners](image)

  - **Observations:**
    - MAX-MAX and MIN-MIN process can cover other corners
    - Unimodal representation gives overly pessimistic min and max delay values, especially as CD mean difference between groups increases
    - Skewed process (MAX-MIN or MIN-MAX for G1-G2) can induce large delay variation across color assignments
Path Delay of DPL (2)

- **Normalized delay variations (Sigma / Mean)**
  - Mean and sigma of a long inverter chain (16 stages) over all process corners (Min and Max combinations)
  - Alternatively colored paths have smaller delay variation → this is a design optimization!
Electrical Impact on Timing Slack

- **Timing slack**

\[ T_{\text{slack}} = T_{\text{clock\_path}} + T_{\text{cycle}} - T_{\text{data\_path}} \]

- **Timing slack variation**

\[ \sigma_{T_{\text{slack}}}^2 = \sigma_{T_{\text{clock\_path}}}^2 + \sigma_{T_{\text{data\_path}}}^2 - 2 \operatorname{cov}(T_{\text{clock\_path}}, T_{\text{data\_path}}) \]

Large correlation is better for the timing slack

- I.e., uncorrelated clock and data path creates large slack variation

- **Clock skew**

- Especially, clock skew from uncorrelated launching and capturing clock paths are a major source of timing slack variation.
Illustration of Timing Analysis

**DPL**
- Path delay variation is small
- No spatial correlation assumed
- Example circuit:
  - Clock path delay and data path delay is same
    - Mean delay is 10ns
  - Variation of the path delays:
    - $\pm 2\text{ns}$
  - Worst slack calculation without spatial correlation

**Single exposure case**
- Path delay variation is large
- Perfect spatial correlation assumed
- Example circuit:
  - Clock path delay and data path delay is same,
    - Mean delay is 10ns
  - Variation of the path delays:
    - $\pm 5\text{ns}$
  - Worst slack calculation with perfect spatial correlation

\[
\text{Worst slack} = \min(\text{clock}) - \max(\text{data})
\]

\[
= 8 - 12 = 4\text{ns}
\]

Worst slack = 5-5=0ns

Worst slack = 15-15=0ns
Simulation Setup for Corner Cases

- **Sample design**
  - AES (aes_cipher_top) from Opencores
  - 45nm Libraries from Nangate
  - 45nm bulk CMOS SPICE model from ASU’s PTM

- **Extracted the most critical path in AES**
  * Exhaustive tests (4 x 2^{54}) are not feasible

- Data path: 30 stages
- Clock launch: 14 stages
- Clock capture: 14 stages
Clock Skew of DPL

- **Assumption:** Data path coloring is fixed
- **Clock path configuration**

<table>
<thead>
<tr>
<th>Case</th>
<th>Launch</th>
<th>Capture</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M12+M12...</td>
<td>M12+M12...</td>
</tr>
<tr>
<td>2</td>
<td>M21+M21...</td>
<td>M21+M21...</td>
</tr>
<tr>
<td>3</td>
<td>M12+M12...</td>
<td>M21+M21...</td>
</tr>
<tr>
<td>4</td>
<td>M21+M21...</td>
<td>M12+M12...</td>
</tr>
<tr>
<td>5</td>
<td>M12+M21...</td>
<td>M12+M21...</td>
</tr>
</tbody>
</table>

- **Even for the zero mean difference case, clock skew exists and increases with mean difference**
  - Pooled unimodal cannot distinguish this clock skew
- **Implication for clock network design**
  - Use the same colored cells for both clock paths
  - Alternatively colored path can further reduce variation

Max: 53.4 ps @ 6nm difference
Max: 22.7 ps @ 0nm difference
Timing Slack of DPL

- An originally zero slack path turns out to have significant negative slack (red line)

Timing slack (s) for MAX-MAX combination

- Pooled unimodal shows very pessimistic slack (but, still can not see the delay difference from different coloring → can not cover all timing problems, such as hold violations)
DPL Analysis Implications Summary

- **Observations**
  - Bimodal CD distribution in DPL increases within-die variation → Increases design difficulty
  - Bimodal-aware timing analysis and optimization are necessary

- **Research directions**
  - Clock path design to reduce clock skew variation
  - Data path design to reduce worst-case delay and delay variation
  - Flip-flop design to improve robustness over the bimodal process corners
## Conclusion

- **Research performed by:**
  - Kwangok Jeong
  - Chul-Hong Park
  - Dr. Hailong Yao

- **IMPACT member company feedback**
  - When we start IMPACT research at UCSD, will DPL enablement (e.g., as scoped here) be of interest?
  - What particular directions (cell layout, auto-placement, timing model characterization, variability-minimizing optimizations, …) are of greatest interest?
  - What collaboration and validation mechanisms?
Thank You!