Physical design and mask synthesis considerations for DPT

Kevin Lucas, Chris Cork, John Hapli, Alex Miloslavsky
Synopsys

Vincent Wiaux, Staf Verhaegen
IMEC
Double Patterning – pitch splitting

1st trench imaging
- Resist
- BARC
- HM
- LowK
- Etch stop
- ....

Hard Mask etch
Resist/BARC strip

2nd trench imaging

Hard Mask etch
Resist/BARC strip

transfer to dielectric
Outline

• EDA areas for DPT
• Full-custom layout
• Automatic Physical Design
• Tapeout & Mask Synthesis
• Conclusions
## Double Patterning in EDA Flow

<table>
<thead>
<tr>
<th>EDA area</th>
<th>Goals</th>
</tr>
</thead>
</table>
| Full custom layout               | Minimize effort & area  
Maximize electrical performance |
| Std cell creation & characterization | Minimize cell width  
Maximize electrical performance |
| Place & route                    | Maximize routability (=time/effort)  
Minimize area |
| Tapeout signoff (DRC)            | Guarantee no violations in full layout  
Fast |
| Mask Synthesis                   | Yield friendly  
Fast |
22nm logic layers: DPT or SE+RDR?

- Estimated from layout review & previous gen methods
- Active: single expose + RDR? DPT?
  - Relatively few small spaces need DPT except in SRAM
- Poly: L-E + subtractive L-E + restricted pitch
  - SRAM & logic line-end cutting
  - More benefit from subtractive L-E cut than from DPT?

B. Arnold IEDM ’06
- also TI public website
22nm logic layers: DPT or SE+RDR?

- **Contact:** DPT + RDR (gridded?)
  - Must allow contacted poly pitch & redundant contacts
- **Metal1:** DPT
  - Big area benefit of DPT + layout creativity (but tough for design)
Full custom DPT layout
Smart coloring for custom layout

Least critical errors returned as violations to save effort/area

More printable coloring = blue
- Blue: diagonal spacing is too small → non-compliant
- Red: horizontal spacing is smaller → less-compliant

Designers also need ability to ensure cutting/coloring done a specific way - memory cells, transistor matching, density uniformity, etc.

EDA flow must enable transfer of designer specified cut & color info
Automated DPT design flows
DPT routing overview

- Questionable if required for 22nm node logic
- Successful implementation in automatic P&R flow appears difficult (SPIE 6924-02, 2008)
  - Traditional design rules not sufficient
  - Flow of route-detect errors-repair shown possible but concerned about high #s of cases where jumpers or rip-up & re-route required.
Predictable Success

DPT standard cell generation flow

- DPT Decompression Tool
- External analysis
- Error Reports
- GDS Import
- Tune Layout
- Corrected Layout
- GDS Layout

FixRules: 1, 2, ..., N

Good hints are key
DPT Correction Flow
Example: aoi22x1 with 2 METAL1 conflicts

Source GDS

GDS analyzed for conflicts

2 Conflicts on METAL1 found

Pass 1 Conflicts corrected and cell optimized

Corrected GDS 1 New Conflict

Pass 2 Conflicts corrected and cell optimized

Corrected GDS

POLY & METAL1 DPT Analysis Results
DPT standard cell library migration

• Took a traditional standard cell library and converted to a DPT clean library for poly and metal1 layers
• ~150 unique standard cells
• Compared cell area before and after conversion
• Results:
  • Area ~same after conversion (!?)
• 2 possible reasons
  • A) DPT avoids halation rules min space OK
  • B) Compaction engine improvements since library created
• Initial conclusion: Area increase is small inside Std cells
  • To be confirmed with same compaction engine
DPT friendly Std cell boundaries

No guarantee that DPT friendly std cells create DPT friendly layout after placement. Ques: can we design placement friendly std cells efficiently (effort & area used)?
DPT friendly placement of Std Cells

• Observations: placed Std cells are guaranteed DPT friendly if:
  - Min space to cell boundary is $\geq \frac{1}{2}$ min single expose space,
  - OR
  - Feature of width $\approx 2\times$min CD straddles boundary,
  - OR
  - Boundary features are gridded and only have single CD (parallel to the boundary)
  - OR
  - Only single color within $\frac{1}{2}$ min single expose space to boundary,
DPT friendly placed M1 std cell design

1. Brute force method: add extra width to each cell
   - simple, fast, guaranteed DPT friendly
   - std cell area impact: 3-7% (unacceptably high)

2. Only single color < ½ min single expose space to bndry
   - implemented in existing prototype std cell creation flow
   - 1st placed DPT friendly std cell ‘library’ created
   - area impact appears small (but to be confirmed)

Dummy lines to enforce DPT clean at cell left/right during layout optimization
DPT Tapeout & Mask Synthesis
DPT Tapeout/Decomposition/OPC

• Functions:
  - decomposition (split & color), RET, OPC and error identification.

• High yield
  - Split creates OPC/RET friendly polygons
  - OPC creates yield friendly wafer patterns
    - through overlay, focus, & dose variations
  - Very high accuracy/predictive OPC models
  - Symmetry, density uniformity

• Fast turnaround
  - Limits complexity/iterations in decomposition algorithms

• **Must** successfully convert all DPT “compliant” designs
Model-based DPT in Mask Synthesis
- Can use OPC model to predict and reduce errors

Looks fine after split but has error on wafer

Looks bad after split but is good on wafer

MRC limits OPC at LE

DPT-aware OPC
Summary and Conclusions

• DPT EDA flow has multiple requirements
  ▪ Custom, std cell, routing, tapeout, mask synthesis
• Not all 22nm logic layers expected to need DPT
• Custom DPT layout possible today
  ▪ Intelligent methods reduce effort & performance variation
• DPT automated routing difficult & need uncertain
• DPT-friendly std cell creation required & demonstrated
  ▪ DPT friendly inside std cells and between standard cells
  ▪ Fix guidance improves automation
• Fast, error-proof, model-based full chip DPT required for tapeout & mask synthesis
Acknowledgements

• Ben Painter, Martin Drapeau, Brian Ward, Stephen Jang, Levi Barnes, Gerry Luk-Pat: Synopsys
• Mireille Maenhoudt, Eric Hendrickx: IMEC
• Will Conley: Freescale
• James Blatchford, Tom Aton, Scott Jessen: TI
Backup foils:
Triple patterning algorithms
- all foils from Chris Cork, PMJ08
## Design Shrink: 4 Contact Cell

<table>
<thead>
<tr>
<th></th>
<th>Design Layout</th>
<th>Split Layout</th>
<th>Design pitch : Splitting pitch ratio</th>
<th>Area</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Layer:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Square Grid</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
|                  | ![Design Layout](image) | ![Split Layout](image) | x = 1  
y = 1                        | 100%  |
| Double Patterning: |               |              |                                       |       |
| Rectangular Grid |               |              |                                       |       |
| Grid → Hexagonal | ![Design pitch](image) | ![Splitting pitch](image) | x = \(\sqrt{3}/2\)  
y = 1/2 | 43%   |
| Patterning:      |               |              |                                       |       |
| Hexagonal Packing|               |              |                                       |       |
| Triple Patterning:    |               |              |                                       |       |
| Hexagonal Packing  | ![Design pitch](image) | ![Splitting pitch](image) | x = 1/\(\sqrt{3}\)  
y = 1/2 | 29%   |

- Using different design grids for \(x\) and \(y\) may with multiple patterning may allow significantly greater packing densities.
- Triple patterning at contact complemented by double patterning on \(X\) and \(Y\) bit-line/word line layers may enables a 2 node reduction cell area.
Penrose Tiling

- Penrose Tiling was proposed by the British Mathematician Roger Penrose in 1970.
- A pattern is created using a limited number of tiles, yet completely covers a surface with an infinitely repeatable pattern showing no translation symmetry.
- Rhombus Penrose tiling has 5-fold symmetry and has been shown to be 3-colorable.[1]
- It is also believed to underlie some examples of 800 year old Islamic art, such as this example from the Seljuk Mama Hatun Mosque in Tercan, Turkey.

3-Colored Penrose tiled contact array

- Pitch split Penrose Tiling patterns consists of a network of small odd and even cycles.
- The cycles contain between 3 and 7 contacts each sharing no more than one common edge with any other cycle.
- Considering reflection and rotational symmetry, there are multiple possible solutions.
- Every contact (apart from those at the edge) is linked to 4 other contacts.

5th Generation Penrose Rhombic Tiling showing Multiple odd cycle loops. Requires only 3 colors for compliance.
Scalability Results

<table>
<thead>
<tr>
<th>Generation</th>
<th>Nodes</th>
<th>WCSP (s)</th>
<th>SAT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>30</td>
<td>0.97</td>
<td>&lt;0.01</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>2.27</td>
<td>&lt;0.01</td>
</tr>
<tr>
<td>4</td>
<td>170</td>
<td>7.35</td>
<td>0.02</td>
</tr>
<tr>
<td>5</td>
<td>480</td>
<td>914</td>
<td>0.31</td>
</tr>
<tr>
<td>6</td>
<td>1300</td>
<td>4.86</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>3505</td>
<td>93.1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>9345</td>
<td>1581</td>
<td></td>
</tr>
</tbody>
</table>

- Beyond a critical number of nodes, computational time seems to increase exponentially for all algorithms.
- Smarter algorithms can handle larger networks in a reasonable time.
- This places a limit on the size of networks that can be colored.