A Methodology for the Early Exploration of Design Rules for Multiple-Patterning Technologies

Rani S. Ghaida, Tanaya Sahu, Parag Kulkarni, and Puneet Gupta

rani@ee.ucla.edu

NanoCAD Lab
Motivation

- Multiple-patterning technology inevitable at 20nm (possibly 14nm)
  - Pitch-split double/triple patterning
  - Self-aligned double-patterning
- All have forbidden patterns and impose specific design rules
  → Decomposition conflicts
Outline

- Prior Art – Dealing with Conflicts
- Our Approach
- Experimental Results
- Summary and Future Work
Prior Art – Dealing with Conflicts

Correct-by-construction

- Design with MP rules $\rightarrow$ hassle
- Conservative rules $\rightarrow$ area cost

[TCAD’12]

Post-layout Legalization

e.g. [ICCAD’11]

- Considerable layout modification
- Considerable area increase
- May require manual redesign
This Work – A Hybrid Approach

- Framework to identify design rules that bring MP conflicts down to a manageable number for legalization
- Estimate the layout and predict conflicts from estimation data
  1. Fast device-layers generation
  2. Wiring/congestion estimation
  3. Conflict prediction w/ machine learning
Outline

- Prior Art – Dealing with Conflicts
- **Our Approach**
  - Device-Layers Generation
  - Wiring/Congestion Estimation
  - Machine-Learning-Based MP Conflict Prediction
- Experimental Results
- Summary and Future Work
DRE – Fast Device-Layers Generation

- Designed for fast and early exploration of design rules
- Supports latest device-level technologies: finFET and local interconnects
- Generates device layers for standard cells in seconds

**Used at industry**
150 rules/styles
20 process params

http://nanocad.ee.ucla.edu/Main/DownloadForm
Outline

- Prior Art – Dealing with Conflicts
- Our Approach
  - Device-Layers Generation
  - Wiring/Congestion Estimation
- Machine Learning-Based MP Conflict Prediction
- Area Vs. MP Compatibility

Trial Design Rules

Trial MP Rules

Device-Layers Generation

Wiring/Congestion Estimation

Machine Learning-Based MP Conflict Prediction

Area Vs. MP Compatibility
Why Wiring Estimation NOT Actual Routing?

• **Infinitely many solutions** → Different tools/designers/design effort can reach completely different solutions
  – Cannot generalize/trust results for tech development
• **Long run-time** → not suitable for exploring wide range of rules
• **Unavailability of tools** supporting MP techs
Probabilistic Layout and Congestion Estimation

- Resembling routing-level probabilistic congestion estimation
- Grid with **configurable tile size** of fixed horz/vert # of tracks
- **Enumerate possible wiring solutions** in the net’s bounding box following a single-trunk Steiner tree topology
  - Each solution will have a **probability of occurrence**

*Probability of each wiring solution for this net is 1/6*
Enumeration of possible wiring solutions

- For nets with very small bounding box
  - Too few solutions, one in the extreme case
    → **Expand bounding box** (i.e. allow detours)
- For nets with skewed aspect ratio
  - Unrealistically long wiring in one direction
    → **Ignore solutions w/ trunks along shorter direction**
Congestion Estimation

- Update congestion in each tile
  - Congestion = Occupied / Available track length
  - Occupied = Utilization + Blocked Regions
  - Utilization = Wire-length X probability of occurrence
- Design-rules violations → blocked regions
- Overflow in tiles → increase cell area & update congestion
- Report tile congestion and features distribution
Validation of Device-Layer Estimation

- Compare DRE estimates with area of actual std-cells
- For Nangate Open Cell Library (110 cells)
  - Less than 2% error on average
  - 38 minutes for entire library on single CPU
Validation of Layout Estimation

- Good accuracy identifying tiles with high congestion
- Good wire-length estimates compared with FLUTE rectilinear Steiner-tree router on entire library
  - 12% higher wire-length on average
  - 44X faster
Prior Art – Dealing with Conflicts

Our Approach
- Device-Layers Generation
- Wiring/Congestion Estimation
- Machine-Learning-Based Conflict Prediction

Experimental Results
Summary and Future Work
Training/Testing The Conflict Predictor

- Netlist
- Actual Process Design Rules
- Decomposition Rules
- Actual Layouts

DRE w/ Probabilistic Layout Estimation

Machine-Learning Model for Conflict-Prediction

Golden Conflict-Detection

Compare

Training Feedback / Testing Results
DP Conflict Prediction w/ Machine Learning

- Neural-networks-based conflict predictor in layouts
- Input layout descriptors
  - Horizontal/vertical tile congestion
  - # and locations of tips, L and T-shapes
- Model supports 3 spacing rules with different values (S2S, T2S, T2T)

<table>
<thead>
<tr>
<th>Property</th>
<th>ANN Model</th>
</tr>
</thead>
<tbody>
<tr>
<td># of layers / # of neurons</td>
<td>3 / 12</td>
</tr>
<tr>
<td># of inputs</td>
<td>19</td>
</tr>
<tr>
<td>Supported min same-color spacings</td>
<td>S2S, T2S, T2T</td>
</tr>
<tr>
<td>Training error</td>
<td>8.7%</td>
</tr>
</tbody>
</table>
Account for Conflicts at Cell-Boundaries

- Conflicts may occur at interface b/w cells post placement
  - Include interface layouts from benchmark designs in training and testing
Predictor’s Testing Results

- Positive detection rate 82.5%
- Negative detection rates 80%
Outline

- Prior Art – Dealing with Conflicts
- Our Approach
  - Device-Layers Generation
  - Wiring/Congestion Estimation
  - Machine-Learning-Based Conflict Prediction
- Experimental Results
- Summary and Future Work
Exploration Setup

- DP-compatibility metric

\[ \text{Design Conflicts} = \alpha \times CC + (1 - \alpha) \times CB \]

- CC → # of conflicting cells in library
- CB → # of conflicting cell-boundary snippets in design
- \( \alpha \) → weighting factor

- Results for 4 benchmark designs (1.5 to 20K cell designs)
- Nangate 45nm standard-cell library at M1 layer
- FreePDK 45nm design rules with scaled decomposition rules
- Baseline experiment:
  - 10-track cell height, 1D poly, Local interconnects to perform poly-to-poly connections
Exploring Tip-to-Side Same-Color Rule

Min spacing = 65nm

- Increase rule value → increase in design conflicts
  - Increase from 1.7 to 2x min spacing → 2X design conflicts
  - Pushing rule value in high-sensitivity range → big benefits
Exploring Tip-to-Side Design Rule

- Non-linear trend $\rightarrow$ optimization opportunities
  - Change rule from 65nm to 80nm $\rightarrow$ 3% less design conflicts cells with almost no area overhead
  - Change to 100nm $\rightarrow$ 9% less design conflicts and 4.7% area increase

Min spacing = 65nm
Min same-color spacing = 130nm
Impact of Cell-Height

• Cell-height \(\rightarrow\) considerable impact on DP conflicts
• DP-compatibility change correlated with area change
Impact of Local Interconnects

- Area and DP-compatibility almost **insensitive to LI scheme**
Summary

• Decomposition conflicts $\rightarrow$ one of the biggest DP challenges
• A hybrid approach
  – Explore rule options that will bring down conflicts to a manageable number
• First work on early exploration of DP rules for speeding up rules-development cycle
• Able to identify rule-ranges with high sensitivity to area and DP-compatibility

• Method applicable for SADP and TP
• Restricting specific patterns in future work
Thank you for your attention!

Questions?