Stitch Aware Detailed Placement for Multiple E-Beam Lithography

Yibo Lin¹, Bei Yu², Yi Zou¹,³, Zhuo Li⁴, Charles J. Alpert⁴, and David Z. Pan¹

¹ECE Department, University of Texas at Austin
²CSE Department, Chinese University of Hong Kong
³CEAS Department, Nanjing University
⁴Cadenace Design Systems, Inc.

This work is supported in part by NSF and SRC
Outline

- Introduction
- Previous Work
- Problem Formulation
- Stitch Aware Detailed Placement
- Experimental Results
- Conclusion
### Technology Scaling

<table>
<thead>
<tr>
<th>Exposure tool</th>
<th>Pitch</th>
</tr>
</thead>
<tbody>
<tr>
<td>Immersion</td>
<td>80</td>
</tr>
<tr>
<td></td>
<td>76</td>
</tr>
<tr>
<td></td>
<td>72</td>
</tr>
<tr>
<td></td>
<td>66</td>
</tr>
<tr>
<td></td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>48</td>
</tr>
<tr>
<td></td>
<td>44</td>
</tr>
<tr>
<td></td>
<td>40</td>
</tr>
<tr>
<td></td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>32</td>
</tr>
<tr>
<td></td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>12</td>
</tr>
<tr>
<td>EUV</td>
<td>5</td>
</tr>
<tr>
<td>ArF, EUV, E-beam</td>
<td>4</td>
</tr>
<tr>
<td>Nanoimprint</td>
<td>1</td>
</tr>
<tr>
<td>High NA EUV</td>
<td>7</td>
</tr>
<tr>
<td>E-beam</td>
<td>8</td>
</tr>
</tbody>
</table>

**Uni-directional parallel line/space patterning techniques**

- Consenses that technique has been used in production
- Published demonstrations from potential deployable equipment show opportunity for production
- Simulations, surface images, or research grade demonstration suggest potential for extendability

[Courtesy ITRS]
E-Beam Lithography

- Direct-write or mask?

[Courtesy E-beam Initiative]
Multiple E-Beam Lithography

- Massively-Parallel e-beam writing
  - Each stripe has width of 50~200 microns
  - Stitching region has a width around 15nm [Berg+, SPIE’11]
  - Field stitching

Stripes

Field

[Fang+, DAC’13] MAPPER Lithography System
Field Stitching

- SEM figures showing stitches at boundaries of beam stripes
Previous Work

- Stitch aware routing for MEBL
  - [Fang+, DAC’13], [Liu+, TCAD’15]
- TPL aware placement
  - [Yu+, TCAD’15], [Kuang+, TVLSI’15], [Chien+, TCAD’15]
  - [Tian+, ICCAD’14], [Lin+, ISPD’15]
  - TPL applies different constraint to placement from MEBL
- No placement algorithm addressing MEBL stitch constraint yet
Stitch Errors

• Defects on vias and vertical wires

• Defects on short polygons

[Fang+, DAC2013]
Stitch Errors within Standard Cell

Resolve stitch errors by proper placement
Dangerous Site Representation

- A cell is divided into sites (poly pitch)
- Sites that contain susceptible segments are marked as “dangerous sites”
Problem Formulation

● Input
  • Initial placement
  • Dangerous site information for each standard cell (precomputed)

● Output
  • New placement with optimized wirelength and minimum stitch errors
  • MEBL friendliness
Given a set of ordered cells $c_1, c_2, \ldots, c_n$, place cells horizontally to minimize objectives such as wirelength or movement.

Previous work on single row algorithm

- Conventional objectives
  - [Brenner+, DATE’00], [Kahng+, GLSVLSI’04], Abacus
    [Spindler+, ISPD’08], [Taghavi+, ICCAD’10]

- TPL awareness
  - [Yu+, ICCAD’13]: $O(mnK)$
  - [Kuang+, ICCAD’14]

Note: $\tau = 10$, $\phi = 1$, $v = 1$ in the experiment
Single Row Placement

- Given a set of ordered cells $c_1, c_2, \ldots, c_n$, with maximum cell displacement $M$
  - Minimize wirelength and stitch errors
  - An algorithm supports a cost function generalizes wirelength, movement and stitch errors

$$cost_i(p_i) = \tau \cdot WL(p_i) + \phi \cdot MOV(p_i) + \nu \cdot SP(p_i)$$

Note: $\tau = 10, \phi = 1, \nu = 1$ in the experiment
Single Row Placement

- Given a set of ordered cells $c_1, c_2, \ldots, c_n$, with maximum cell displacement $M$
  - Shortest path solved by dynamic programming
  - $O(nM^2)$
**Speedup with Pruning Techniques**

- **Pruning technique 1**
  - Let $t_i(p_i)$ denote the cost of placement solution from $c_1$ to $c_i$ in which $c_i$ is placed at $p_i$.
  - Comparing two solutions $\alpha_i(p_i)$ and $\alpha_i(q_i)$, if $t_i(p_i) \geq t_i(q_i)$ and $p_i \geq q_i$, then $\alpha_i(p_i)$ is inferior to $\alpha_i(q_i)$.
  - Prune inferior solutions

\[
\begin{align*}
\text{Solution } \alpha_i(p_i) & \quad \text{Solution } \alpha_i(q_i) \\
\begin{array}{c}
C_i \\
p_i
\end{array} & \quad \begin{array}{c}
C_i+1 \\
p_i+1
\end{array} \\
\begin{array}{c}
C_i \\
q_i
\end{array} & \quad \begin{array}{c}
C_i+1 \\
q_i+1
\end{array}
\end{align*}
\]

Value sets of $p_{i+1}$ and $q_{i+1}$
Speedup with Pruning Techniques

• Pruning technique 2
  • Let $p_{i-1}^*$ be the optimal position of cell $c_{i-1}$ when cell $c_i$ is placed at $p_i$
  • Let $q_{i-1}^*$ be the optimal position of cell $c_{i-1}$ when cell $c_i$ is placed at $q_i$
  • If $q_i \geq p_i$, then $q_{i-1}^* \geq p_{i-1}^*$
  • Reduce searching ranges

Value sets of $p_{i-1}$ and $q_{i-1}$

\[\text{Solution } \alpha_i(p_i)\]

\[\text{Solution } \alpha_i(q_i)\]
Effectiveness of Speedup Techniques

- **$O(nM)$ complexity**
  - Requirements: $cost_i(p_i)$ only depends on $p_i$
  - 30x speedup
  - Keep optimality
Resolve Stitch Errors in Dense Regions

- Global swap to smooth out density
  - \( \text{score}(c_i, c_j) = \Delta s\text{HPWL} - \lambda \cdot P_{ds} - \mu \cdot P_{ov} \)

Overlap penalty

sHPWL change

Normalized penalty of dangerous site density

\[
P_{ds} = \max(0, |D'_{ds}(i) - D'_{ds}(j)| - |D_{ds}(i) - D_{ds}(j)|) \cdot A_b
\]

\( D_{ds}(i) \): the density of dangerous sites in bin \( B_i \) before swap

\( D'_{ds}(i) \): the density of dangerous sites in bin \( B_i \) after swap

\( A_b \): bin area

Achieve better density of dangerous sites

Note: \( s\text{HPWL} = HPWL \times (1 + \alpha \times P_{ABU}) \) from ICCAD 2013 Contest
Overall Flow

Initial Placement

Stitch Aware
Single Row Placement

Zero Stitch Errors?

Output Placement

Stitch Aware Global Swap

N

Y
Experimental Environment Setup

- Implemented in C++
- 8-Core 3.4GHz Linux server with 32GB RAM
- ICCAD 2014 contest benchmark
  - Mapped to Nangate 15nm Standard Cell Library
  - Legalized with RippleDP [Chow+, ISPD’14]

<table>
<thead>
<tr>
<th>Design</th>
<th>#cells</th>
<th>#nets</th>
<th>#blockages</th>
</tr>
</thead>
<tbody>
<tr>
<td>vga_lcd</td>
<td>165K</td>
<td>165K</td>
<td>0</td>
</tr>
<tr>
<td>b19</td>
<td>219K</td>
<td>219K</td>
<td>0</td>
</tr>
<tr>
<td>leon3mp</td>
<td>649K</td>
<td>649K</td>
<td>0</td>
</tr>
<tr>
<td>leon2</td>
<td>794K</td>
<td>795K</td>
<td>0</td>
</tr>
<tr>
<td>mgc_edit_dist</td>
<td>131K</td>
<td>133K</td>
<td>13</td>
</tr>
<tr>
<td>mgc_matrix_mult</td>
<td>155K</td>
<td>159K</td>
<td>16</td>
</tr>
<tr>
<td>netcard</td>
<td>959K</td>
<td>961K</td>
<td>12</td>
</tr>
</tbody>
</table>
Experimental Results

Wirelength Improvement %

Final Stitch Errors

Init.: initial input placement
SR: single row algorithm only
Full Flow: apply full flow including single row algorithm and global swap
Runtime Comparison

- Full flow is slightly slower than SR
  - Only apply to regions still containing stitch errors

Runtime (s)
Conclusion

• Methodology to handle e-beam stitch errors during detailed placement stage

• A linear time single row algorithm with highly-adaptable objective functions

• Can be utilized in existing CAD tool on optimizing: Wire-length; Routability; Congestion, etc.

• Future work
  • Consider interaction between placement and routing for EBL friendliness
Thanks