A Multilevel Analytical Placement for 3D ICs

Jason Cong$^{1,2}$ & Guojie Luo$^1$
$^1$UCLA Computer Science Department
$^2$California NanoSystems Institute
{ cong, gluo } @ cs.ucla.edu

This work is partially supported by IBM (under a DARPA subcontract) and NSF
3D Integration

Example (MIT Lincoln Lab 180nm SOI technology)

- A collection of *tiers*
- Through-silicon via (*TSV*)
Basic 3D Placement Problem

◆ Variables
  ▪ \((x_i, y_i, z_i), i=1,2,...,n\)
  ▪ cell \(i\) is placed at \((x_i, y_i)\) on the tier \(z_i\)

◆ Objective
  ▪ \(\sum_e WL_e(x,y,z) = HPWL_{(x,y)} + \alpha_{TSV} HPWL_z\)
  ▪ To minimize weighted wirelength

◆ Constraint
  ▪ no overlap between cells
Previous Works on 3D Placement

- Force-directed method
  - [Goplen & Sapatnekar, ICCAD’03]

- Partitioning-based method
  - [Goplen & Sapatnekar, DAC’07]

- Quadratic modeling of density cost through DCT
  - [Yan et al., Integration’09]

- 2D to 3D transformation method
  - [Cong et al., ASPDAC’07]
**Motivations**

- **3D placement tool**
  - Trade-offs between wirelength and TSV
  - Flexible to integrate other objective function and constraints
  - High-quality and scalable

- To study analytical placement
Our Contributions

- Analytical formulation with a novel density penalty function
  - Based on multiple-tier 2D density penalty functions
  - Introduce pseudo-layers, so that minimization of penalties on tiers and pseudo tiers guarantees a legal 3D placement

- Adaption of multilevel method
  - Provides extra TSV reduction in addition to increasing the TSV weight

- Improvements compared to 2D to 3D transformation
  - (best wirelength cases) 2% shorter wirelength and 29% fewer TSV
  - (best TS via cases) 20% shorter wirelength and 50% fewer TSV
Basic 3D Placement Problem

◆ Variables
  ▪ \((x_i, y_i, z_i), \ i=1,2,...,n\)
  ▪ cell \(i\) is placed at \((x_i, y_i)\) on the tier \(z_i\)

◆ Objective
  ▪ \(\sum_e W_{Le}(x,y,z) = HPWL_{(x,y)} + \alpha_{TSV} HPWL_z\)
  ▪ To minimize weighted wirelength

◆ Constraint
  ▪ no overlap between cells
**Weighted Wirelength**

\[ WL_e(x, y, z) = \left( \max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) + \alpha_{TSV} \cdot \max_{v_i, v_j \in e} |z_i - z_j| \]
Weighted Wirelength

\[ WL_e(x, y, z) = \max_{v_i, v_j \in e} \left( |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) \]

2D HPWL
Weighted Wirelength

\[ WL_e(x, y, z) = \left( \max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) + \alpha_{TSV} \cdot \max_{v_i, v_j \in e} |z_i - z_j| \]

3D Weighted HPWL
**Weighted Wirelength**

\[
WL_e(x, y, z) = \left( \max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) + \alpha_{TSV} \cdot \max_{v_i, v_j \in e} |z_i - z_j|
\]

- Model TSV by a length of wire
  - For example [Davis et al., DTC’05]
    - MIT Lincoln Lab 180 nm 3D SOI technology
    - 3 \( \mu \)m thick TSV \( \approx \) 8 to 20 \( \mu \)m metal 2 wire, in terms of capacitance
    - 3 \( \mu \)m thick TSV \( \approx \) 0.2 \( \mu \)m metal 2 wire, in terms of resistance
Another case

- Tier 1 and tier 2: face-to-face
- Tier 2 and tier 3: back-to-back

Different weights between tiers
Weighted Wirelength

- Practical weighed wirelength

\[
WL_e(x, y, z) = (1 + p_e) \left( \max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j| \right) \\
+ (1 + q_e) \cdot \alpha_{TSV} \cdot \max_{v_i, v_j \in e} |z_i - z_j|
\]

- Additional net weights \( p_e \) and \( q_e \) to model and optimize performance or temperature [Goplen & Sapatnekar, DAC’07]

- It is a convex function w.r.t. \((x, y, z)\)

- Such weighted wirelength is the form of objective function in the 3D placement problem formulation
Analytical Engine

◆ Discrete tier assignment

- Discrete (legalized solution)

◆ Relaxed tier assignment

- Relaxed (intermediate solution)

◆ Variables

- \((x_i, y_i, z_i), \ i = 1, 2, \ldots, n\)
- cell \(i\) is placed at \((x_i, y_i)\) on the tier \(z_i\)
Analytical Engine

◆ Discrete tier assignment

◆ Relaxed tier assignment

◆ Formulate 3D placement problem as continuous optimization

minimize $\sum_e WL_e (x, y, z)$
subject to (no overlap between cells)
Non-overlap Constraints

- Relaxed by area density constraints
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps
    - Cell overlaps in overflow bins violate density constraints
    - Cell overlaps not in overflow bins do not violate density constraints
Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps

minimize \[ \sum_e WL_e (x, y, z) \]
subject to (no overlap between cells)
\[ \sum_e WL_e (x, y, z) \leq C_{i,j,k} \]
for all \( i, j, k \)
**Non-overlap Constraint**

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps

\[
\begin{align*}
\text{minimize} & \quad \sum_{e} W_{L_e} (x, y, z) \\
\text{subject to} & \quad \text{no overlap between cells} \\
\text{minimize} & \quad \sum_{e} W_{L_e} (x, y, z) \\
\text{subject to} & \quad A_{i,j,k} (x, y, z) = C_{i,j,k} \\
& \quad \text{for all } i, j, k
\end{align*}
\]

[Chan et al., ISPD’06]
Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps

\[
\begin{align*}
\text{minimize} & \quad \sum_e WL_e (x, y, z) \\
\text{subject to} & \quad A_{i,j,k} (x, y, z) = C_{i,j,k} \quad \text{for all } i, j, k \\
\end{align*}
\]

\[
\begin{align*}
\text{minimize} & \quad \sum_e WL_e (x, y, z) + \frac{\mu}{2} \sum_k \sum_{i,j} (A_{i,j,k} (x, y, z) - C_{i,j,k})^2 \\
\text{increase } \mu \text{ until overlaps are removed} & \\
\end{align*}
\]

[Nam & Cong, Springer’07]
[Cong & Luo, ISPD’08]
Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps

- Area projection to obtain bin densities from intermediate solution
Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps
- Area projection to obtain bin densities from intermediate solution
Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps

- Area projection to obtain bin densities from intermediate solution
Non-overlap Constraint

- Replaced by area density constraint
  - Divide the placement region into bins
  - Measure the overflow of bin area to capture cell overlaps

- Area projection to obtain bin densities from intermediate solution
Area Projection

Bell-shaped function to project area

\[ \eta(k, z) = \begin{cases} 
1 - 2(z - k)^2 & |z - k| \leq 1/2 \\
2(|z - k| - 1)^2 & 1/2 < |z - k| \leq 1 \\
0 & \text{otherwise}
\end{cases} \]

\( \eta(k, z) \) - The projection ratio from “tier z” to tier k

\[
A_{i, j, k}(x, y, z) = \sum_{v \in V} A_{i, j}(x_v, y_v) \cdot \eta(k, z_v)
\]
**Area Projection**

**Bell-shaped function to project area**

\[ \eta(k, z) = \begin{cases} 
1 - 2(z - k)^2 & |z - k| \leq 1/2 \\
2(|z - k| - 1)^2 & 1/2 < |z - k| \leq 1 \\
0 & \text{otherwise} 
\end{cases} \]

**An Example**

- Intermediate placement of a cell at “tier 2.316”
- Projects 0% area to tier 1
- Projects 80% area to tier 2
- Projects 20% area to tier 3
- Projects 0% area to tier 4
Area Projection

Bell-shaped function to project area

\[ \eta(k, z) = \begin{cases} 
1 - 2(z - k)^2 & |z - k| \leq 1/2 \\
2(|z - k| - 1)^2 & 1/2 < |z - k| \leq 1 \\
0 & \text{otherwise}
\end{cases} \]

\( \eta(k, z) \) - The projection ratio from “tier z” to tier k

An Example

- Intermediate placement of a cell at “tier 2.316”
- Projects 0% area to tier 1
- Projects 80% area to tier 2
- Projects 20% area to tier 3
- Projects 0% area to tier 4
Equivalence to Non-overlap Constraint

- Area projection to tiers is not enough
  - Counter example: projected area failed to capture illegality
  - Solution: area projection on pseudo-tiers
Equivalence to Non-overlap Constraint

◆ Theorem: \((x, y, z)\) satisfy the constraints

\[
\begin{align*}
A_{i,j,k}(x, y, z) &= C_{i,j,k} \\
A'_{i,j,k}(x, y, z) &= C'_{i,j,k}
\end{align*}
\]

for all \(i, j, k\)

if.f. \((x, y, z)\) is a legal placement (no overlaps)

** after adding filler cells
Equivalence to Non-overlap Constraint

**Theorem:** \((x, y, z)\) is a minimizer of the function:

\[
\frac{\mu}{2} \sum_k \sum_{i,j} \left( A_{i,j,k}(x, y, z) - C_{i,j,k} \right)^2
\]

\[
+ \frac{\mu}{2} \sum_k \sum_{i,j} \left( A'_{i,j,k}(x, y, z) - C'_{i,j,k} \right)^2
\]

if.f. \((x, y, z)\) is a legal placement (no overlaps)

**after adding filler cells**
minimize \[ \sum_e WL_e (x, y, z) \]
\[ + \frac{\mu}{2} \sum_k \sum_{i,j} (A_{i,j,k}(x, y, z) - C_{i,j,k})^2 \]
\[ + \frac{\mu}{2} \sum_k \sum_{i,j} (A'_{i,j,k}(x, y, z) - C'_{i,j,k})^2 \]

increase \( \mu \) until overlaps are removed

- \( A_{i,j,k}(x, y, z) \): area projected in bin \((i,j)\) of tier \(k\)
- \( C_{i,j,k} \): area capacitance on tier \(k\)
- \( A'_{i,j,k}(x, y, z) \): area projected in bin \((i,j)\) of pseudo-tier \(k\)
- \( C'_{i,j,k} \): area capacitance on pseudo-tier \(k\)
Multilevel Framework

Level at which analytical engine is applied

- C  Coarsening
- I  Interpolation
Experimental Results (1/2)

Comparison of trade-off curves (ibm13)

- 19% shorter WL
- 9% fewer TSV
- 15% shorter WL
- 43% fewer TSV

(consistent behavior on other circuits)
## Experimental Results (2/2)

- **The ability to reduce the TSV number**

<table>
<thead>
<tr>
<th>Circuit</th>
<th>3-Level Placement</th>
<th>4-way Mincut</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GP WL (x 10^7)</td>
<td>DP WL (x 10^7)</td>
</tr>
<tr>
<td>ibm01</td>
<td>0.39</td>
<td>0.39</td>
</tr>
<tr>
<td>ibm03</td>
<td>0.92</td>
<td>0.91</td>
</tr>
<tr>
<td>ibm04</td>
<td>1.36</td>
<td>1.31</td>
</tr>
<tr>
<td>ibm06</td>
<td>1.67</td>
<td>1.62</td>
</tr>
<tr>
<td>ibm07</td>
<td>2.79</td>
<td>2.70</td>
</tr>
<tr>
<td>ibm08</td>
<td>2.99</td>
<td>2.89</td>
</tr>
<tr>
<td>ibm09</td>
<td>2.36</td>
<td>2.29</td>
</tr>
<tr>
<td>ibm13</td>
<td>5.02</td>
<td>4.89</td>
</tr>
<tr>
<td>ibm15</td>
<td>12.05</td>
<td>11.40</td>
</tr>
<tr>
<td>ibm18</td>
<td>18.36</td>
<td>17.37</td>
</tr>
<tr>
<td>geo-mean</td>
<td>2.66</td>
<td>2.58</td>
</tr>
</tbody>
</table>
Summary

- Non-overlap constraints
  - Handled by a novel area projection method
  - Pseudo-tiers added for equivalence to non-overlap constraints

- Multilevel framework
  - Effective to reduce TS via number

- Trade-offs between WL and #TSV
  - 12% shorter WL and 29% fewer TSV
    - Compared to the 2D to 3D transformation method with best WL
  - 20% shorter WL and 50% fewer TSV
    - Compared to the 2D to 3D transformation method with best TSV
Thank you!