Low-Power Gated Bus Synthesis for 3D IC via Rectilinear Shortest-Path Steiner Graph

Chung-Kuan Cheng, Peng Du, Andrew B. Kahng, and Shih-Hung Weng
UC San Diego
Email: ckcheng@ucsd.edu
Outline

• Introduction
• Statement of Problem
• Algorithms
  – Determination of TSV locations
  – Generating Rectilinear Shortest-Path Steiner Graph
• Experimental Results
• Conclusion
Introduction: 2D bus

- Problem: a gated bus with multiplexers and demultiplexers to minimize power consumption
- Shorest-Path Steiner Graph: a graph that contains shortest paths between sources and sinks, with minimal total wire length
Introduction: 3D Bus

• Through Silicon Vias (TSV) for inter-silicon connection
  – Silicon area
  – Feature size
  – Yield

• Implication:
  – The z segment is more expensive than x & y segments
  – Routing distance between different layers may not be the shortest
Statement of Problem

• Given: A set of masters ($src$) and a set of slaves ($dst$) on $L$ silicon layers, and traffic demands between all ($src$, $dst$) pairs

• Assumption: time sharing bus, one channel on each direction. Routing is optimized and fixed.

• Objective: (1) Power consumed by the traffic and (2) total wire length

• Output: 3D Steiner graph

• Constraint: bounded #TSVs one each silicon layer
Motivational Example

• **src:** s1, s2, **dst:** t1, t2
• Traffic Demands:
  - \( (s1, t1) = 5, (s1, t2) = 1 \)
  - \( (s2, t1) = 3, (s2, t2) = 4 \)
• \#TSV/layer= 1
• Wire length
  - \( (2+5+1)+(1+3+5) \)
• Power consumption
  - \( 5 \times 7 + 1 \times 7 + 3 \times 11 + 4 \times 9 \)

One channel for each direction
Power = demand \times length
Overall Flow

Locations of masters and slaves on device layers

Communication frequencies between each master-slave pair

Determine TSV Position

TSV position of each layer

Generate Shortest-Path Steiner Graph per Device Layer

Shortest-Path Steiner Graph
Problem Formulation

- **TSV Placement**: Place TSVs between adjacent layers so that the total traffic power (length of weighted shortest paths between src-dst pairs) is minimized.

- **Steiner Graph on Each Layer**: Given a silicon layer $k$ with TSV locations on both sides, construct a shortest-path Steiner graph to connect all traffics between srcs, dsts, and TSVs on layer $k$. 
TSV Placement (#TSV/layer=1)

- For #TSV=1, we can decompose 2D placement into 1D.

- A dynamic programming algorithm is proposed to find optimal TSV locations.
  - Let $Opt(k,r)$ be the minimal total traffic power among terminals $(src, dst)$ in the first $k$ layers and the TSV between layers $k$ and $k+1$ at location $r$.

- Algorithm complexity is $O((n+m)^2L)$, where $n=$ #srcs, $m=$ #dsts, $L=$ #layers.
TSV Placement (#TSV/layer>1)

1. Snap the Hanan points into a coarse grid, e.g. 5x5

2. Find the best TSV placement on the snapped Hanan points using exhaustive search

3. For every TSV, refine the placement.

4. Repeat step 3 until there is no improvement.
Steiner Graph on Each Layer (tree merge)

1. Start with $m$ dsts as $m$ trees. Each root of the tree contains an src list to be connected.

2. Merge a pair of roots $p$ and $q$ with the largest benefit. Update the src list on the new root.

3. Repeat step 2 until there is no more pairs to be merged.

4. For the roots of nonempty src list, route to the srcs on the list.

5. Remove redundant edges.

Computational Complexity $O(nm^2)$
Steiner Graph on Each Layer (tree merge)

- Our objective is to connect each one of $s_1, s_2, s_3, s_4, s_5$ to $p$ and $q$.
- By merging $p$ and $q$, the benefit is the total length of blue segments.
Steiner Graph on Each Layer (LP Rounding)

- The figure depicts the directed network $N_l$ on the Hanan grid.
- The rectilinear shortest path from $s^l$ to $t^l$ corresponds to a flow with amount one in $N_l$. 

$S^l$ is below $t^l$

$S^l$ is above $t^l$
Steiner Graph on Each Layer (LP Rounding)

\[
\begin{align*}
\text{min} & \quad \sum_{(u,v) \in E_H} d_{uv} x_{uv} \\
\text{s.t.} & \quad \sum_{(u,v) \in E_l} f^l_{uv} - \sum_{(v,u) \in E_l} f^l_{vu} \geq 0, \\
& \quad 1 \leq l \leq Q \text{ and } v \neq s^l, t^l; \\
& \quad \sum_{(s^l, u) \in E_l} f^l_{su} \geq 1, \quad 1 \leq l \leq Q; \\
& \quad \sum_{(u, t^l) \in E_l} f^l_{ut} \geq 1, \quad 1 \leq l \leq Q; \\
& \quad x_{uv} - f^l_{uv} \geq 0, \\
& \quad (u, v) \in E_H \text{ and } l \in \{l_0 : (u, v) \in E_{l_0}\}; \\
& \quad f^l_{uv} \in \{0, 1\}, \quad 1 \leq l \leq Q \text{ and } (u, v) \in E_l; \\
& \quad x_{uv} \in \{0, 1\}, \quad (u, v) \in E_H.
\end{align*}
\]

- \( E_h \): undirected edge set of Hanan grid.
- \( E_l \): directed edge set on top of \( E_h \) for each demand \( l \)
- \( f^l_{u,v} \): flow from \( u \) to \( v \) on edge \((u,v)\) in \( E_l \).
- \( Q \): \# demands (src, dst)
- \( x \): a binary variable to denote the selection of edge \((u,v)\) in the graph.
- \( d \): wire length of edge \((u,v)\).
Steiner Graph on Each Layer (LP Rounding)

• Solve the LP relaxation of the ILP formulation.
• Sort the edges with respect to the decreasing order of the x variables.
• Delete edges as long as the remaining graph contains necessary shortest paths.

#variables: $O((n+m)^2Q)$
Experimental Results (#TSV/layer=1)

The same communication frequencies for all master-slave pairs.

(src, dst) pairs in first two layers communicate 5 times freq.
Experimental Results

#TSV/layer=1
Power=439

#TSVs/layer=2
Power=395

#TSVs/layer=3
Power=348
### Experimental Results: Power

<table>
<thead>
<tr>
<th>((L, N))</th>
<th>(B = 1)</th>
<th>(B = 2)</th>
<th>(B = 3)</th>
<th>(B = \infty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3,10)</td>
<td>456.46</td>
<td>375.93</td>
<td>360.86</td>
<td>346.36</td>
</tr>
<tr>
<td></td>
<td>(31.79%)</td>
<td>(8.54%)</td>
<td>(4.19%)</td>
<td></td>
</tr>
<tr>
<td>(3,20)</td>
<td>384.34</td>
<td>336.13</td>
<td>305.50</td>
<td>292.39</td>
</tr>
<tr>
<td></td>
<td>(31.45%)</td>
<td>(14.96%)</td>
<td>(4.49%)</td>
<td></td>
</tr>
<tr>
<td>(3,50)</td>
<td>453.09</td>
<td>393.07</td>
<td>355.22</td>
<td>337.27</td>
</tr>
<tr>
<td></td>
<td>(34.34%)</td>
<td>(16.55%)</td>
<td>(5.32%)</td>
<td></td>
</tr>
<tr>
<td>(4,20)</td>
<td>476.15</td>
<td>413.28</td>
<td>367.62</td>
<td>343.93</td>
</tr>
<tr>
<td></td>
<td>(38.45%)</td>
<td>(20.16%)</td>
<td>(6.89%)</td>
<td></td>
</tr>
<tr>
<td>(5,20)</td>
<td>492.00</td>
<td>431.04</td>
<td>367.39</td>
<td>346.68</td>
</tr>
<tr>
<td></td>
<td>(41.92%)</td>
<td>(24.33%)</td>
<td>(5.97%)</td>
<td></td>
</tr>
<tr>
<td>(5,50)</td>
<td>452.50</td>
<td>401.23</td>
<td>346.31</td>
<td>323.76</td>
</tr>
<tr>
<td></td>
<td>(39.77%)</td>
<td>(23.93%)</td>
<td>(6.96%)</td>
<td></td>
</tr>
</tbody>
</table>

- \((L,N)\): (# layers, # masters and slaves in each layer)
- \(B\): #TSVs/layer
Experimental Results (Steiner Graph)

Length=6006, 5.38% extra extra
Tree merge

Length=5683, 0% extra extra
LP relaxation and rounding
Experimental Results (Steiner Graph)

<table>
<thead>
<tr>
<th>(n, m)</th>
<th>Previous</th>
<th>Greedy</th>
<th>LP(Obj)</th>
<th>LP(Round)</th>
<th>Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3, 16)</td>
<td>5388</td>
<td>5068</td>
<td>4882</td>
<td>4882</td>
<td>9.39%</td>
</tr>
<tr>
<td>(5, 15)</td>
<td>7024</td>
<td>6586</td>
<td>6342</td>
<td>6342</td>
<td>9.71%</td>
</tr>
<tr>
<td>(12, 6)</td>
<td>6698</td>
<td>6306</td>
<td>5915</td>
<td>5915</td>
<td>11.69%</td>
</tr>
<tr>
<td>(6, 12)</td>
<td>7127</td>
<td>6160</td>
<td>5575</td>
<td>5575</td>
<td>21.78%</td>
</tr>
<tr>
<td>(12, 12)</td>
<td>11559</td>
<td>10385</td>
<td>10319</td>
<td>10319</td>
<td>10.73%</td>
</tr>
<tr>
<td>(20, 20)</td>
<td>16236</td>
<td>15921</td>
<td>13847</td>
<td>13847</td>
<td>14.71%</td>
</tr>
<tr>
<td>(30, 30)</td>
<td>302619</td>
<td>287042</td>
<td>251841</td>
<td>251968</td>
<td>16.78%</td>
</tr>
</tbody>
</table>

Lengths of LP(Obj) and LP(Round) are almost the same with 1.0005 ratio on the last case

• Previous: [Wang DAC09]
• Greedy: Tree merge
• Improvement: Previous vs VP(Round)
CPU Time of LP Relaxation and Rounding

<table>
<thead>
<tr>
<th>$(n,m)$</th>
<th># Variables</th>
<th># Constraints</th>
<th>Time</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>(3, 16)</td>
<td>4326</td>
<td>6209</td>
<td>&lt;1s</td>
<td>&lt;50MB</td>
</tr>
<tr>
<td>(5, 15)</td>
<td>8019</td>
<td>11458</td>
<td>&lt;1s</td>
<td>&lt;50MB</td>
</tr>
<tr>
<td>(20, 20)</td>
<td>155396</td>
<td>239037</td>
<td>3m43s</td>
<td>234MB</td>
</tr>
<tr>
<td>(30, 30)</td>
<td>778358</td>
<td>1175974</td>
<td>4h22m</td>
<td>1.2GB</td>
</tr>
</tbody>
</table>

CPU: Intel Core i3, 2.4GHz; Memory: 4GB
Conclusion

• A framework and algorithms to synthesize the gated bus in 3D ICs.
• Optimal TSV placement when #TSV/layer=1
  Exhaustive search on coarse grid + iterative improvement when #TSV/layer>1
• New Steiner graph algorithms with total wire length reduction of up to 22%.
• Future Works
  – Multiple Path Graph
  – Control Systems
Thank you for your attention!