A Fast Crosstalk- and Performance-Driven Multilevel Routing System

Tsung-Yi Ho, Yao-Wen Chang, Sao-Jie Chen

Graduate Institute of Electronics Engineering
Department of Electrical Engineering
National Taiwan University
Taiwan

D. T. Lee

Institute of Information Science
Academia Sinica
Taiwan
Agenda

• Introduction
• Multilevel Routing Framework
• Experimental Results
• Conclusions
Agenda

• Introduction
• **Multilevel Routing Framework**
• Experimental Results
• Conclusions
Routing Trends

- Billions of transistors may be fabricated in a single chip with nanometer technology.
- Need tools for very large-scale designs.
- Framework evolution for CAD tools:
Two-Stage Routing

- **Global routing**
  - Partition the routing area into tiles.
  - Find tile-to-tile paths for all nets.
  - Optimize given objectives.

- **Detailed routing**
  - Assign actual tracks and vias for nets.
Flat and Hierarchical Routing Frameworks

**Flat**

Drawback: hard to handle larger designs

**Hierarchical**

Drawback: lack the global information for the interaction among subregions
Multilevel Framework

- It has been successfully applied to partitioning, floorplanning, placement and routing in VLSI physical design and many more.

- Ingredients:
  - Bottom-up Coarsening: Iteratively groups a set of circuit components.
  - Top-down Uncoarsening: Iteratively ungroups clustered components and refines the solution.
First Multilevel Router

  - Coarsening: Routing resource estimation
  - Initial global routing: Multicommodity flow algorithm
  - Uncoarsening: Refinement using constrained maze global routing algorithm
Routability- and Performance-Driven Multilevel Router

  - This framework integrates global routing, detailed routing and resource estimation together at each level.

Perform global and detailed routing for local connections and then estimate routing congestion for the next level.

Use maze routing to reroute failed nets and refine the solution.
Our Multilevel Framework

- An intermediate step (layer/track assignment) between coarsening and uncoarsening stage is introduced to do runtime and crosstalk optimization.

Perform congestion-driven pattern routing for local connections and then estimate routing congestion for the next level.

Use point-to-path maze routing to reroute failed nets level by level.

Perform crosstalk-driven layer/track assignment for long segments, and short segments are routed by a point-to-path maze router.
# Multilevel Routing Framework Comparison

<table>
<thead>
<tr>
<th></th>
<th>Coarsening stage</th>
<th>Initial routing</th>
<th>Uncoarsening stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cong et al. in ICCAD 01</td>
<td>Resource estimation</td>
<td>Multicommodity flow</td>
<td>Global maze refinement</td>
</tr>
<tr>
<td>Lin and Chang in ICCAD 02</td>
<td>Global routing</td>
<td>No initial routing</td>
<td>Global and detailed maze refinement</td>
</tr>
<tr>
<td>Our Framework</td>
<td>Global routing</td>
<td>Track/layer assignment</td>
<td>Global and detailed maze refinement</td>
</tr>
<tr>
<td></td>
<td>Resource estimation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Multilevel Routing Framework Comparison

<table>
<thead>
<tr>
<th></th>
<th>Coarsening stage</th>
<th>Initial routing</th>
<th>Uncoarsening stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cong et al. in ICCAD 01</td>
<td>Resource estimation</td>
<td>Multicommodity flow</td>
<td>Global maze refinement</td>
</tr>
<tr>
<td>Lin and Chang in ICCAD 02</td>
<td>Global routing</td>
<td>No initial routing</td>
<td>Global and detailed maze refinement</td>
</tr>
<tr>
<td>Our Framework</td>
<td>Global routing</td>
<td>Track/layer assignment</td>
<td>Global and detailed maze refinement</td>
</tr>
</tbody>
</table>

- **Cong et al.’s framework**
  - Focus on global routing
  - The precise relative positions of nets are not determined
  - Insufficient information for addressing nanometer electrical effects (e.g., crosstalk)
# Multilevel Routing Framework Comparison

<table>
<thead>
<tr>
<th></th>
<th>Coarsening stage</th>
<th>Initial routing</th>
<th>Uncoarsening stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cong et al. in ICCAD 01</td>
<td>Resource estimation</td>
<td>Multicommodity flow</td>
<td>Global maze refinement</td>
</tr>
<tr>
<td>Lin and Chang in ICCAD 02</td>
<td>Global routing</td>
<td>No initial routing</td>
<td>Global and detailed maze refinement</td>
</tr>
<tr>
<td>Our Framework</td>
<td>Global routing</td>
<td>Track/layer assignment</td>
<td>Global and detailed maze refinement</td>
</tr>
</tbody>
</table>

- **Lin & Chang’s framework**
  - Focus on routability-driven routing
  - Most of nets are detailed routed (fixed)
  - Not flexible to address nanometer electrical effects
# Multilevel Routing Framework Comparison

<table>
<thead>
<tr>
<th></th>
<th>Coarsening stage</th>
<th>Initial routing</th>
<th>Uncoarsening stage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cong et al. in ICCAD 01</td>
<td>Resource estimation</td>
<td>Multicommodity flow</td>
<td>Global maze refinement</td>
</tr>
<tr>
<td>Lin and Chang in ICCAD 02</td>
<td>Global routing</td>
<td>No initial routing</td>
<td>Global and detailed maze refinement</td>
</tr>
<tr>
<td>Our Framework</td>
<td>Global routing</td>
<td>Layer/Track assignment</td>
<td>Global and detailed maze refinement</td>
</tr>
<tr>
<td></td>
<td>Resource estimation</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Our new framework**
  - Fully utilizes the information of global router
  - Nets are routed in parallel
  - Suitable to address nanometer electrical effects
Agenda

- Introduction
- Multilevel Routing Framework
- Experimental Results
- Conclusions

Performance-Driven Routing Tree Construction

Crosstalk-Driven Layer/Track Assignment
Performance-Driven Routing Tree

- **Minimum Spanning Tree (MST)**
  - Has the smallest total wire length
  - May incur longer path length
- **Shortest Path Tree (SPT)**
  - Has the shortest path length
  - May incur larger total wire length

Minimum Radius Minimum Cost Spanning Tree (MRMCST) is proposed for performance-driven routing tree construction.
Union of All MSTs

- Construct union and intersection graphs of all MSTs to preserve the minimum cost.
Union of All MSTs

- Construct **union** and **intersection** graphs of all MSTs to preserve the minimum cost.

**Time complexity:** $O(n \log n)$

**How to choose the right optional edges?**
Preprocessing

• Every blue component should keep
  ▪ Pseudo-center \((pc)\): center of the longest path
  ▪ Distance of every node to its \(pc\)
  ▪ Radius from \(pc\)
  ▪ Optional edges incident to the blue component
Locally Optimal Connection Strategy (LOCS)

- LOCS: Choose an edge \( e = (p, q) \) to minimize the cost

\[
 f(e, T) = \text{dist}(s, p) + \text{cost}(e) + \text{dist}(q, pc(T)) + R_T
\]

- Prim-like algorithm

- \( O(n + m_{opt} \log m_{opt}) \)

Number of nodes needed to update the distance from the source \( s \)

Use a heap structure to choose optional edges
Routing Model

- Partition a chip into tiles
- *Multilevel routing graph* $G = (V, E)$
  - Each node in $V$ denotes a tile
  - Each edge in $E$ denotes the boundary of adjacent tiles
Global Routing

• Pattern routing
  ▪ Uses L-shaped and Z-shaped connections to route nets.
  ▪ Has lower time complexity than maze routing.
Resource Estimation

- Global routing cost is the summation of congestions of all routed edges.
- The congestion, $C_e$, of an edge $e$ is defined by

$$C_e = \frac{1}{2^{(p_e - d_e)}},$$

where $p_e$ and $d_e$ are the capacity and density, respectively.
- We update the congestion of routed edges to guide the subsequent global routing.
Agenda

- Introduction
- Multilevel Routing Framework
- Experimental Results
- Conclusions

Performance-Driven Routing Tree Construction

Crosstalk-Driven Layer/Track Assignment
Coupling Capacitance Induced Crosstalk

- Coupling capacitance is more significant than self capacitance in deep submicron era
  - Devices and wires are placed in closer proximity
  - Wire heights scale at a slower rate than widths
- Coupling capacitance is proportional to coupling wire length and inversely proportional to spacing distance.

Coupling can account for up to 70% of interconnect capacitance even in .25 micron designs. # Sylvester et al, in Proc. VLSI Symposium on Technology, 1998.
Coupling Capacitance Induced Crosstalk

- Crosstalk
  - Caused by coupling capacitance
  - Leads to functional failure and wire delay

- Global routing
  - Precise net positions are not determined yet and thus only some sort of estimation can be done.

- Detailed routing
  - Most nets are fixed and thus it is not flexible to handle the wire coupling

A Desirable Intermediate Step:
Layer/Track Assignment
Crosstalk-Driven Layer Assignment

To avoid two segments with larger overlap adjacent to each other on the same layer, we need to separate them to different layers.

Graph k-coloring heuristic on MST to separate segments to k layers

Maximum spanning tree heuristic to extract edges with larger cost
Crosstalk-Driven Track Assignment

Extract the maximum clique in the interval graph

Interval graph

Bipartite assignment graph

Combined graph
Crosstalk-Driven Track Assignment

Select the segment with the maximum degree in the clique as the starting point.

Interval graph + Bipartite assignment graph = Combined graph
Crosstalk-Driven Track Assignment

Remove edges corresponding to segments that can be allocated on track 1 but are overlapped with segment b
Crosstalk-Driven Track Assignment

Assign the remaining segments in the clique to tracks

Interval graph

Bipartite assignment graph

Combined graph
Choose the next maximum clique for track assignment till all segments are assigned.
Our Multilevel Framework Recap

Perform congestion-driven pattern routing for local connections and then estimate routing congestion for the next level.

Perform crosstalk-driven layer/track assignment for long segments, and short segments are routed by a point-to-path maze router.

Use point-to-path maze routing to reroute failed nets level by level.
Agenda

• Introduction
• Multilevel Routing Framework
• Experimental Results
• Conclusions
Experimental Settings

- **Language:** C++
- **Platform:** 1GHz Sun Blade 2000 with 1GB memory
- **Library:** STL, LEDA, LayoutDB (UCLA)
- **Benchmarks taken from previous works:**

<table>
<thead>
<tr>
<th>Circuits</th>
<th>Size (μm)</th>
<th>#Layer</th>
<th>#Nets</th>
<th>#Pins</th>
</tr>
</thead>
<tbody>
<tr>
<td>S5378</td>
<td>4330x2370</td>
<td>3</td>
<td>3124</td>
<td>4734</td>
</tr>
<tr>
<td>S9234</td>
<td>4020x2230</td>
<td>3</td>
<td>2774</td>
<td>4185</td>
</tr>
<tr>
<td>S13207</td>
<td>6590x3640</td>
<td>3</td>
<td>6995</td>
<td>10562</td>
</tr>
<tr>
<td>S15850</td>
<td>7040x3880</td>
<td>3</td>
<td>8321</td>
<td>12566</td>
</tr>
<tr>
<td>S38417</td>
<td>11430x6180</td>
<td>3</td>
<td>21035</td>
<td>32210</td>
</tr>
<tr>
<td>S38584</td>
<td>12940x6710</td>
<td>3</td>
<td>28177</td>
<td>42589</td>
</tr>
</tbody>
</table>
Routability and Running Time

- Obtained about the same routability
- Achieved about 6.7X speedup, used 2X memory

<table>
<thead>
<tr>
<th>Circuits</th>
<th>Results of Lin and Chang (ICCAD 2002)</th>
<th>Our Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (s)</td>
<td>Memory (MB)</td>
</tr>
<tr>
<td>S5378</td>
<td>35</td>
<td>18</td>
</tr>
<tr>
<td>S9234</td>
<td>26.2</td>
<td>14</td>
</tr>
<tr>
<td>S13207</td>
<td>106.7</td>
<td>24</td>
</tr>
<tr>
<td>S15850</td>
<td>538.8</td>
<td>40</td>
</tr>
<tr>
<td>S38417</td>
<td>899.9</td>
<td>75</td>
</tr>
<tr>
<td>S38584</td>
<td>1953.7</td>
<td>496</td>
</tr>
<tr>
<td>Avg</td>
<td>6.7</td>
<td>1</td>
</tr>
</tbody>
</table>
# Delay and Crosstalk

<table>
<thead>
<tr>
<th>Circuits</th>
<th>Results of Lin and Chang (ICCAD 2002)</th>
<th>Our Results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(D_{\text{max}})</td>
<td>(D_{\text{avg}})</td>
</tr>
<tr>
<td>S5378</td>
<td>37308</td>
<td>1403</td>
</tr>
<tr>
<td>S9234</td>
<td>25512</td>
<td>1072</td>
</tr>
<tr>
<td>S13207</td>
<td>55337</td>
<td>1262</td>
</tr>
<tr>
<td>S15850</td>
<td>76297</td>
<td>1302</td>
</tr>
<tr>
<td>S38417</td>
<td>121419</td>
<td>1170</td>
</tr>
<tr>
<td>S38584</td>
<td>150936</td>
<td>1208</td>
</tr>
<tr>
<td>\text{avg}</td>
<td>1.2</td>
<td>1.1</td>
</tr>
</tbody>
</table>

- \(D_{\text{max}}\): Critical path delay (ps)
- \(D_{\text{avg}}\): Average net delay (ps)
- \(C_{\text{max}}\): Maximum coupling length (\(\mu\))
- \(C_{\text{avg}}\): Average coupling length (\(\mu\))

15% Coupling capacitance is reduced about 15%. 
5% not included in delay computation for fair comparison.

30% 
24%
Routing of s5378

Example of a routing result with GUI (s5378)
Agenda

• Introduction
• Multilevel Routing Framework
• Experimental Results
• Conclusions
Conclusions

• We have proposed a novel multilevel routing framework that is suitable to address nanometer electrical effects.

• Compared with the state-of-the-art previous work, our approach achieved a 6.7X runtime speedup, reduced the respective maximum and average crosstalk by about 30% and 24%, reduced the respective maximum and average delay by about 15% and 5%.

• Future work lies in multilevel routing considering other nanometer electrical effects such as antenna effect.
Thank You!!