RegularRoute: An Efficient Detailed Router with Regular Routing Patterns

Yanheng Zhang and Chris Chu
Electrical and Computer Engineering Department
Iowa State University
Outline

- Motivation and Overview
- Local Net Routing
- Global Segment Assignment
- Experimental Results
- Conclusion and Discussion
Outline

- Motivation and Overview
- Local Net Routing
- Global Segment Assignment
- Experimental Results
- Conclusion and Discussion
Previous Detailed Routing Techniques

- **Iterative ripup and reroute**
  - Mighty [Shin et al. TCAD-87] (Sequential in nature)

- **Multi-level methodology**
  - DUNE [Cong et al. TCAD-01]
  - MR [Chang et al. TCAD-04] (Net ordering issue)

- **Boolean satisfiability**
  - SAT Router for FPGA [Nam et al. TCAD-02] (Concurrent approach, Long runtime)

- **Track routing**
  - Track Routing [Batterywala et al. ICCAD-02] (Pin access issue)

- **Escape routing**
  - Escape Routing for Pin Clusters [Ozdal TCAD-09] (Not handle full-chip routing)
Apply Regular Routing Patterns

- Regular routing patterns
  - Potentially improve design rule satisfaction
  - Explore solution space more efficiently
  - Might affect routability due to restricted routing patterns

Non-trivial routing patterns

Regular routing patterns
Problem Formulation for Detailed Routing

- **Input**
  - 3-D detailed routing grids
  - 2-D global routing solution organized in global segments
  - Complete netlist

- **Objective**
  - Generate detailed routing solution to route as many nets as possible
  - Secondary objectives include minimizing wirelength, via count and non-preferred usage

- **Assumptions**
  - Each grid edge can accommodate exact one wire except blockage
  - Each layer has *preferred routing direction*. They are perpendicular for adjacent layers. Metal_1 is assumed to be *horizontal*
  - Pins are assumed to be on metal_1
RegularRoute: Flow and Overview

Input 2-D Global Routing Solution and 3-D Grids

Local Nets Routing by Single Trunk V Tree

Global Segment Extraction

Global Segment Assignment

Output Detailed Routing Solution

Unassigned segments
Assigned segments
RegularRoute: Our Contributions

- Applying regular routing patterns
  - Use regular routing patterns instead of non-trivial patterns
  - Correct-by-construction for satisfying more design rules

- Panel based global segments allocation
  - Formulate assigning global segments in one panel as MWIS problem
  - All nets inside each panel are considered simultaneously

- Novel techniques to improve routability
  - Effective partial assignment for further assignment
  - Pin promotion to prevent pin access issue

- Fast computational time
  - Fast heuristic in solving the MWIS
  - Can easily be adapted to parallel version
Outline

- Motivation and Overview
- **Local Net Routing**
- Global Segment Assignment
- Experimental Results
- Conclusion and Discussion
Local Net Routing

Input 2-D Global Routing Solution and 3-D Grids

Local Net Routing by Single Trunk V-Tree

Global Segment Extraction

Global Segment Assignment

Output Detailed Routing Solution
Single-Trunk V-Tree

- Find pin with median X coordinate
- Construct trunk with vertical wire (metal_2)
- Connect other pins to trunk as branch
- Time complexity: O(n)
V-Tree vs. Arbitrary Tree

- **V-Tree vs. Arbitrary Tree**
  - Number of blocked horizontal tracks: \( V\)-Tree = Arbitrary Tree
  - Number of blocked vertical tracks: \( V\)-Tree < Arbitrary Tree

Minimize metal\_2 usage

Single Trunk V-Tree

Arbitrary Tree
Outline

- Motivation and Overview
- Local Net Routing
- **Global Segment Assignment**
- Experimental Results
- Conclusion and Discussion
Global Segment Assignment

Input 2-D Global Routing Solution and 3-D Grids

Local Nets Routing by Single Trunk V-Tree

Global Segment Extraction

Global Segment Assignment

Solve Global Segment Assignment for all panels

Partial Assignment

Terminal Promotion

Next layer

From 1st layer

Top layer?

Yes

Panel Merging and Maze Routing

No

Output Detailed Routing Solution
Global Segment Assignment in one Panel

- **Input**
  - A set of global segments that have not been assigned
  - A set of routing tracks inside one panel

- **Objective**
  - Assign as many segments as possible in regular routing patterns
  - Minimize wirelength, via count, non-preferred usage

- **Concepts**
  - **Track**: A sequence of grids in preferred routing direction
  - **Panel**: A collection of tracks in one column/row of G-Cells
Concept of a Choice

- **Choice**
  - A valid regular routing solution for one segment
  - Number of choice reflects the flexibility of assignment for one segment
  - Two choices that cannot co-exist cause a *conflict*

![Diagram showing a choice conflict]

conflict
MWIS problem

- Maximum Weighted Independent Set (MWIS)
  - Formulate Global Segment Assignment in one Panel as MWIS problem
  - Introduce conflict graph $G$ with vertex set $V$ and edge set $E$: each vertex represents one choice, each edge represents conflict between two choices
  - Each vertex is assigned a weight representing assignment priority
  - Objective: find the independent set of vertices to maximize total weight
Example of Conflict Graph

conflict

clique
Calculate weight for vertices

\[ W(v) = L - \alpha_1 \times \|R\| + \alpha_2 \times \text{AvD} \]
\[ + \alpha_3 \times (F_1 + F_2) \]

- Contains five components
  - Segment length (number of spanned G-Cells)
  - Terminal connection
  - G-Cell boundary density
  - Flexibility component for ending G-Cell with pending segment
Solve MWIS

\[ C(v) = W(v) - \beta \times i\_deg(v) - \gamma \times o\_deg(v) \]

- Solve MWIS problem
  - Rank vertices based on cost
  - Extract vertex with largest weight and do assignment
  - Update incident vertices and in/out degrees
  - Use *heap* for efficient extraction and update
Partial Assignment

- Improve resource utilization after MWIS
- Assign partial segment starting from terminals
- Post-processing after MWIS
Terminal Promotion

- Terminal connection issue: segment is assigned in upper layer while terminals are on lower layers
- Promote terminals after processing current layer
- Treat new terminals *as if* they are on upper layer
Unassigned Segments on Top Layer

- **Panel Merging**
  - Allow violation of the input global routing solution
  - Offers more flexibility
  - Can be applied in lower layers

- **Maze Routing**
  - Line probe based maze routing
  - 3-D maze routing

- **Optimal MWIS Solver**
  - Last resort for better solving the problem
  - Generally slow and solution quality is not guaranteed
Outline

- Motivation and Overview
- Local Net Routing
- Global Segment Assignment
- **Experimental Results**
- Conclusion and Discussion
Experimental Set-up

- **Testcases**
  - ISPD98 placement benchmark suite derived testcases
  - ISPD05 placement contest benchmark suites derived testcases

- **Computing Platform**
  - 3.16 GHz Intel Xeon processor with 32G memory

- **Input to RegularRoute**
  - Global routing testcases with similar format to ISPD07/08 global routing contest benchmark suites
  - 2-D global routing solutions

- **Input to WROUTE**
  - LEF/DEF design for placed testcases
Flow for making testcases

Input Placement Benchmark

Placer (Dragon, FastPlace 3.1)

Placed Testcases

In-house script

Global routing testcases similar to ISPD07/08

Global Router (FastRoute)

2-D global routing solutions

RegularRoute

Publicly available conversion tool
PlaceUtil by Umich

LEF/DEF design

WROUTE
## Results for Local Net Routing

<table>
<thead>
<tr>
<th></th>
<th>Single Trunk V-Tree</th>
<th>RSMT</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># Local Nets</td>
<td># un. Local</td>
<td>CPU (Sec.)</td>
</tr>
<tr>
<td>ibm01</td>
<td>1081</td>
<td>0</td>
<td>0.04</td>
</tr>
<tr>
<td>ibm02</td>
<td>1750</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>ibm07</td>
<td>4479</td>
<td>0</td>
<td>0.18</td>
</tr>
<tr>
<td>ibm08</td>
<td>5539</td>
<td>0</td>
<td>0.23</td>
</tr>
<tr>
<td>ibm09</td>
<td>5429</td>
<td>0</td>
<td>0.20</td>
</tr>
<tr>
<td>ibm10</td>
<td>2984</td>
<td>0</td>
<td>0.27</td>
</tr>
<tr>
<td>ibm11</td>
<td>6983</td>
<td>0</td>
<td>0.26</td>
</tr>
<tr>
<td>ibm12</td>
<td>2433</td>
<td>0</td>
<td>0.32</td>
</tr>
</tbody>
</table>
# Full Results for ISPD98 Derived Testcases

<table>
<thead>
<tr>
<th></th>
<th>FR4.0</th>
<th>RegularRoute</th>
<th>WROUTE(Encounter)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU (Sec.)</td>
<td># un. assign</td>
<td>CPU (Sec.)</td>
</tr>
<tr>
<td>ibm01</td>
<td>0.47</td>
<td>0</td>
<td>3.17</td>
</tr>
<tr>
<td>ibm02</td>
<td>2.71</td>
<td>0</td>
<td>14.4</td>
</tr>
<tr>
<td>ibm07</td>
<td>8.51</td>
<td>0</td>
<td>34.3</td>
</tr>
<tr>
<td>ibm08</td>
<td>10.1</td>
<td>0</td>
<td>54.6</td>
</tr>
<tr>
<td>ibm09</td>
<td>6.11</td>
<td>0</td>
<td>43.1</td>
</tr>
<tr>
<td>ibm10</td>
<td>8.97</td>
<td>0</td>
<td>66.9</td>
</tr>
<tr>
<td>ibm11</td>
<td>15.7</td>
<td>0</td>
<td>68.1</td>
</tr>
<tr>
<td>ibm12</td>
<td>25.4</td>
<td>0</td>
<td>112.1</td>
</tr>
<tr>
<td>Sum</td>
<td>77.97</td>
<td>0</td>
<td>396.7</td>
</tr>
<tr>
<td>Norm</td>
<td>0.25</td>
<td>/</td>
<td>1</td>
</tr>
</tbody>
</table>
## Full Results for ISPD05 Derived Testcases

<table>
<thead>
<tr>
<th></th>
<th>FR4.0</th>
<th>RegularRoute</th>
<th>WROUTE (Encounter)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>CPU (Sec.)</td>
<td># un. assign</td>
<td>CPU (Sec.)</td>
</tr>
<tr>
<td>a1</td>
<td>141</td>
<td>0</td>
<td>622</td>
</tr>
<tr>
<td>a2</td>
<td>189</td>
<td>0</td>
<td>558</td>
</tr>
<tr>
<td>a3</td>
<td>342</td>
<td>0</td>
<td>1176</td>
</tr>
<tr>
<td>a4</td>
<td>289</td>
<td>4</td>
<td>1330</td>
</tr>
<tr>
<td>b1</td>
<td>134</td>
<td>0</td>
<td>911</td>
</tr>
<tr>
<td>b2</td>
<td>249</td>
<td>0</td>
<td>1177</td>
</tr>
<tr>
<td>Sum</td>
<td>3384</td>
<td>4</td>
<td>5774</td>
</tr>
<tr>
<td>Norm</td>
<td>0.22</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Outline

- Motivation and Overview
- Local Net Routing
- Global Segment Assignment
- Experimental Results
- Conclusion and Discussion
Conclusion and Future Work

**Conclusion**
- We proposed RegularRoute for routing with regular routing patterns in detailed routing
- Propose a layer by layer and panel by panel strategy to solve global segment assignment
- Formulate MWIS and solved by fast heuristic
- Proposed other effective methods for improving QoR

**Future Work**
- Continue improve performance of RegularRoute
- Incorporate more design-related objectives
- Develop parallel version of RegularRoute
Thank You!

Questions?