An Enhanced Global Router
With Consideration of General Layer Directives

Tsung-Hsien Lee\textsuperscript{1}, Yen-Jung Chang\textsuperscript{1}, and Ting-Chi Wang\textsuperscript{2}

\textsuperscript{1}Department of Electrical and Computer Engineering, University of Texas at Austin
\textsuperscript{2}Department of Computer Science, National Tsing Hua University, Taiwan
Agenda

- Background
- Problem formulation
- Previous work: GLADE
- Our router
- Experimental results
- Conclusion
Global Routing

- Global routing determines tile-to-tile routes of nets

- Conventional Metrics
  - Total overflow (TOF)
  - Total wirelength (TWL)

3D Grid Graph for Global Routing
Produced by making some modifications to ISPD 2008 benchmarks

Specifying layer directives for a subset of nets (LD nets)

Layer directive: a range of consecutive layers on which the net should be routed
- Different LD types whose layer ranges have proper subset relations
- Do not support arbitrary layer ranges

[1:4] (not supported)
Problem Formulation

- **Input**: a multi-layer global routing instance with a subset of nets associated with **general layer directives**
  - The two ends of a layer range can be any metal layers
- **Output**: a global routing solution that minimizes
  - total LD violation as well as TOF
    - A LD net passing through an edge on a non-preferred layer causes one unit of LD violation on the edge
  - TWL
Previous Work: GLADE [ICCAD10]

- Handling ICCAD 2009 benchmarks and hence only targeting a restricted set of layer ranges
- Extending NTHU-Route 2.0 [TCAD10] by performing
  - Pseudo layer assignment during 2D routing
  - LD-aware layer assignment
GLADE: Pseudo Layer Assignment

- Exploited during 2D global routing
- Predicting the amount of LD violations that may occur after actual layer assignment, subject to no overflow increase
- Calculating *virtual capacity (VC)* and *virtual demand (VD)* which are also used to define edge costs for LD nets during the iterative ripup-and-reroute process
Illustration of VC (1/4)

- 3D edges $e'_1$, $e'_2$, $e'_3$, $e'_4$ are projected to a 2D edge $e$
- Three LD types: t1, t2, t3
Illustration of VC (2/4)

□ $v_{c_e}(t1) = 5$

\[ e'_4 \begin{array}{c} 5 \\ e'_3 \begin{array}{c} 5 \\ e'_2 \begin{array}{c} 10 \\ e'_1 \begin{array}{c} 10 \end{array} \end{array} \end{array} \begin{array}{c} \{ t1 \} \end{array} \]
Illustration of VC (3/4)

\[ \text{vc}_e(t2) = 5 + 5 = 10 \]
\[ v_{c_e}(t3) = 5 + 5 + 10 = 20 \]
Illustration of VD (1/3)

\[ \text{vd}_e(t_1) = 4 \]
Illustration of VD (2/3)

- $v_{d_e}(t1) = 4$
- $v_{d_e}(t2) = 4 + 3 = 7$

\[ vd(t1) = 4 \]
\[ vd(t2) = 4 + 3 = 7 \]
Illustration of VD (3/3)

- $v_d(t_1) = 4$
- $v_d(t_2) = 4 + 3 = 7$
- $v_d(t_3) = (4 + 1 + 2) + 12 = 19$
LD Overflow (LDOF)

- \( LDOF_e(t) = \max(v_{d_e}(t) - v_{c_e}(t), 0) \)
  - How many LD nets of type \( t \) that pass through \( e \) cannot be assigned to their preferred layers without causing additional overflow

- \( LDOF_e = \sum_t LDOF_e(t) \)

- Total LDOF = \( \sum_e LDOF_e \)

- At each ripup-and-reroute iteration, GLADE tries to minimize TOF and total LDOF
GLADE: Layer Assignment

- Modifying the layer assignment method (COLA) of NTHU-Route 2.0 [TCAD’08]
  - Net ordering
    - LD nets appear before non-LD nets
  - Single-net layer assignment
    - Minimizing via count
    - Considering layer directives by adding penalty to the routing edges of LD nets which are not located in target layer ranges
- Keeping TOF identical to that of the 2D routing result
Our Router

- Enhancing GLADE to handle general layer directives during 2D global routing and layer assignment
  - Modifying the pseudo layer assignment method for calculating virtual demands
  - Adopting two-stage layer assignment without increase in TOF
    - Initial layer assignment for via count minimization
    - Iterative refinement for further minimizing LD violation and via count
Calculation of VD (1/4)

- 3D edges $e'_1$, $e'_2$, $e'_3$ and $e'_4$ are projected to a 2D edge $e$
- We show how to calculate $v_{de}(t5)$
- First, LD types are sorted in a non-decreasing order of the sizes of their layer ranges

$$\frac{1}{4}$$

$e'_4$ 5
$e'_3$ 5
$e'_2$ 10
$e'_1$ 10

$t1$
- demand=2

$t2$
- demand=12

$t3$
- demand=12

$t4$
- demand=3

$t5$
- demand=1
Calculation of VD (2/4)

- **Step 1** (considering $e'_4$ and $e'_1$)
  - Assigning 2 nets of $t1$ and 3 nets of $t4$ to $e'_4$
  - Assigning 10 nets of $t2$ to $e'_1$
Calculation of VD (3/4)

- **Step 2** (considering e’_3 and e’_2)
  - Assigning 5 nets of t3 to e’_3
  - Assigning 2 nets of t2 and 7 nets of t3 to e’_2
We get \( v_{d_e}(t_5) = (5 + 2 + 7 + 10) + 1 = 25 \)
Two-Stage Layer Assignment: Initial Layer Assignment

- Adopting the layer assignment method COLA [TCAD’08] without considering layer directives
  - Targeting via count minimization
  - Keeping TOF identical to that of the 2D result
Two-Stage Layer Assignment: Refinement (1/5)

- Refining the solution for further minimization of LD violation and via count, but without TOF increase
  - Putting all 2D edges into a queue
  - Iteratively dequeuing an edge and applying a min-cost max-flow technique to re-assign its layer
    - If improved, accepting the result and enqueuing neighboring edges (if they are not in the queue)
Two-Stage Layer Assignment: Refinement (2/5)

- 2D edge without overflow

Wire cap. (1,3,5) = (1,0,1)

Net a [1:3]
- Net b [3:5]

Net a
- e₁ = 2
- e₂ = 5
- e₃ = 2
- e₅ = 1

Net b
- e₄ = 4
- e₆ = 4

#Via = 3 + 3 + 3 + 3 = 12
#LD-Vio = 2

#Via = 1 + 1 + 1 + 1 = 4
#LD-Vio = 0
Two-Stage Layer Assignment: Refinement (3/5)

capacity/cost

1/0

Nets

1/0

Layers

1/2

1/2

1/2

1/6+p

1/6+p

1/2

#Via=Cost = 4
#LD-Vio = 0
Two-Stage Layer Assignment: Refinement (4/5)

- 2D edge with overflow

Wire cap. (1, 3, 5) = (0, 0, 1)

Net a [1:3]
- e₁ = 2
- e₂ = 5
- e₃ = 2
- e₄ = 4
- e₅ = 1
- e₆ = 4

Net b [3:5]
- e₁ = 2
- e₂ = 3
- e₃ = 2
- e₄ = 4
- e₅ = 5
- e₆ = 4

Via = 3 + 3 + 3 + 3 = 12
LD-Vio = 2

Via = 1 + 1 + 1 + 1 = 4
LD-Vio = 0
Two-Stage Layer Assignment: Refinement (5/5)

cap./lower-bound flow/cost

Two graphs are shown, each representing a network of nets and layers with connections between them. The graphs illustrate the flow and cost allocation across layers.

The first graph shows a network with connections labeled with flow and cost values, such as 1/0/2, which represents a flow of 1, 0 lower-bound, and a cost of 2.

The second graph, colored in red, shows an alternative path with similar connections, indicating a possible refinement or alternative solution.

Key metrics for this solution are:
- #Via = Cost = 4
- #LD-Vio = 0

These metrics suggest an optimized solution with a specific number of via connections and layer violations.
Experimental Results

- Our router was implemented in C++
- All experiments were conducted on a Linux machine with Intel 2.2Ghz CPU and 8GB RAM
- Compared with two routers
  - GLADE
    - ICCAD 2009 benchmarks
  - NTHU-Route 2.0
    - Modified ICCAD 2009 benchmarks by randomly changing the layer ranges of LD nets
## GLADE vs. Our Router

| Benchmarks | GLADE | | | | | | | Our Router | | | | |
|------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|            | TOF   | LDOF  | LD    | TWL   | CPU   | TOF   | LDOF  | LD    | TWL   | CPU   |
| adaptec1   | 0     | 0     | 0     | 45.4  | 7.0   | 0     | 0     | 59849/0 | 45.2/45.3 | 10.3  |
| adaptec2   | 0     | 0     | 0     | 43.9  | 1.4   | 0     | 0     | 183623/0 | 43.2/43.8 | 4.2   |
| adaptec3   | 0     | 0     | 0     | 115.2 | 7.2   | 0     | 0     | 210387/0 | 115.0/114.9 | 11.3  |
| adaptec4   | 0     | 0     | 0     | 106.5 | 1.8   | 0     | 0     | 283214/0 | 105.9/106.5 | 3.9   |
| adaptec5   | 0     | 0     | 0     | 130.1 | 15.2  | 0     | 0     | 66706/0 | 129.9/129.6 | 26.0  |
| bigblue1   | 0     | 0     | 0     | 48.3  | 8.7   | 0     | 0     | 53858/0 | 48.5/48.5 | 17.1  |
| bigblue2   | 0     | 0     | 0     | 69.6  | 7.0   | 0     | 0     | 7248/0  | 69.6/69.1 | 10.4  |
| bigblue3   | 0     | 0     | 0     | 105.9 | 3.8   | 0     | 0     | 45669/0 | 105.7/105.5 | 10.4  |
| bigblue4   | 188   | 0     | 0     | 178.9 | 121.0 | 188   | 0     | 71248/0 | 178.7/177.6 | 324.8 |
| newblue1   | 2     | 0     | 0     | 35.6  | 4.8   | 2     | 0     | 6314/0  | 35.6/35.5 | 8.7   |
| newblue2   | 0     | 0     | 0     | 59.7  | 0.8   | 0     | 0     | 49218/0 | 59.5/59.6 | 2.4   |
| newblue4   | 140   | 0     | 0     | 108.1 | 40.1  | 140   | 0     | 45643/0 | 107.9/107.7 | 48.6  |
| newblue5   | 0     | 0     | 0     | 190.7 | 12.6  | 0     | 0     | 9031/0  | 190.7/190.3 | 20.8  |
| newblue6   | 0     | 0     | 0     | 139.8 | 11.5  | 0     | 0     | 26887/0 | 139.8/139.0 | 23.7  |
| newblue7   | 78    | 0     | 0     | 281.7 | 119.9 | 78    | 0     | 113369/0 | 281.2/279/3 | 169.9 |
| Comp.      | --    | --    | 1.000 | 1.000 | 1.000 | --    | --    | --/1.000 | 0.998/0.996 | 1.904 |
## NTHU-Route 2.0 vs. Our Router

<table>
<thead>
<tr>
<th>Benchmarks</th>
<th>NTHU-Route 2.0</th>
<th>Our Router</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TOF</td>
<td>LD Vio</td>
</tr>
<tr>
<td>adaptec1</td>
<td>0</td>
<td>95066</td>
</tr>
<tr>
<td>adaptec2</td>
<td>0</td>
<td>289132</td>
</tr>
<tr>
<td>adaptec3</td>
<td>0</td>
<td>394924</td>
</tr>
<tr>
<td>adaptec4</td>
<td>0</td>
<td>440412</td>
</tr>
<tr>
<td>adaptec5</td>
<td>0</td>
<td>120402</td>
</tr>
<tr>
<td>bigblue1</td>
<td>0</td>
<td>139562</td>
</tr>
<tr>
<td>bigblue2</td>
<td>0</td>
<td>23070</td>
</tr>
<tr>
<td>bigblue3</td>
<td>0</td>
<td>101772</td>
</tr>
<tr>
<td>bigblue4</td>
<td>162</td>
<td>130542</td>
</tr>
<tr>
<td>newblue1</td>
<td>0</td>
<td>13224</td>
</tr>
<tr>
<td>newblue2</td>
<td>0</td>
<td>90746</td>
</tr>
<tr>
<td>newblue4</td>
<td>138</td>
<td>73450</td>
</tr>
<tr>
<td>newblue5</td>
<td>0</td>
<td>36910</td>
</tr>
<tr>
<td>newblue6</td>
<td>0</td>
<td>36276</td>
</tr>
<tr>
<td>newblue7</td>
<td>62</td>
<td>174794</td>
</tr>
<tr>
<td>Comp.</td>
<td>--</td>
<td>1.000</td>
</tr>
</tbody>
</table>
Conclusion

- We have presented a global router that enhances a prior work, GLADE, to handle general layer directives.
- Encouraging experimental results have been provided to support our router.
- A possible future work is to improve our router for further reducing overflow values for benchmarks that are currently difficult to route.
THANK YOU

Q&A