MARCH: MAze Routing
Under a Concurrent and Hierarchical Scheme for Buses

Speaker:
Jingsong Chen

Authors:
Jingsong Chen, Ph.D. student, CSE, CUHK
Jinwei Liu, Ph.D. student, CSE, CUHK
Gengjie Chen, Ph.D. student, CSE, CUHK
Dan Zheng, Ph.D. student, CSE, CUHK
Evangeline F. Y. Young, Professor, CSE, CUHK
Outline

• Problem Formulation

• Our Methodology: MARCH

• Experimental Results

• Conclusion
Outline

• Problem Formulation

• Our Methodology: MARCH

• Experimental Results

• Conclusion
Problem Formulation

• Input:
  • A set of routing layers
    • With preferred track direction for each layer
    • With min spacing constraint for each layer
    • With design boundary for each layer
    • With a set of obstacles for each layer
  • A set of routing tracks
    • With wire width constraint for each track
  • A set of buses
    • With a set of bits to be routed on tracks for each bus
    • With width constraint for each layer for each bus
    • With pin shapes for each bit in a bus

*This case comes from the slides “ICCAD 2018 CAD Contest Problem B Summary” (link: http://iccad-contest.org/2018/)
Problem Formulation

• Output:
  • A set of on-track wires and vias that connect pins for all buses
    • The bits in the same bus must share the exactly same topology

*This case comes from the slides “ICCAD 2018 CAD Contest Problem B Summary” (link: http://iccad-contest.org/2018/)

A toy case*
Problem Formulation

• Evaluation rule:
  • Overall cost consists of routing cost and penalty cost
    • Routing cost is the summation of:
      • Wire length cost: shorter -> better
      • Segment cost: less -> better
      • Compactness cost: more compact -> better

*These cases come from the slides “ICCAD 2018 CAD Contest Problem B Summary” (link: http://iccad-contest.org/2018/)
Problem Formulation

- Evaluation rule:
  - Overall cost consists of routing cost and penalty cost
    - Penalty cost is the summation of:
      - Spacing violation penalty
      - Routing failure penalty
        - Wire off-track
        - Track width violation
        - Bit open
        - Topology inconsistency

*This figure comes from the slides “ICCAD 2018 CAD Contest Problem B Summary” (link: [http://iccad-contest.org/2018/](http://iccad-contest.org/2018/))
Outline

• Problem Formulation

• Our Methodology: MARCH

• Experimental Results

• Conclusion
Our Methodology: MARCH

• If routing bit by bit:
  • Advantage: traditional routing methods can be applied naturally.
  • Disadvantage: topology consistency can hardly be maintained in a relatively complex routing environment.
Our Methodology: MARCH

• Two key features of MARCH:
  • Hierarchically: a topology-aware path planning (coarse-grained) and a track assignment for bits (fine-grained).
  
    ➡️ EFFICIENCY

  • Concurrently: route all the bits in a bus concurrently.

    ➡️ CORRECT-BY-CONSTRUCTION
Our Methodology: MARCH

• Data Structures:
  • Bus-based Grid Graph (BGG):
    • A multilayer grid graph with uniform grid cells (G-cells)
    • A row of G-cells, named *frontline*, is the propagation unit in routing process.
    • Each edge stores edge capacity for the bus and history cost.
  • Track occupancy of each track:
    • Record the positions of the track segments which cannot be used since:
      • Occupied by obstacles
      • Occupied by the routed wires of the other buses
Our Methodology: MARCH

• The overall flow of MARCH:
  • Initialization
    • Construct a BGG with empty edge capacities
  • Inner loop for each bus \( b \):
    • Update the BGG for \( b \):
      • Edge capacities meeting width constraint
      • Pin locations
    • Generate a routing path consisting of a set of rectangular regions in topology-aware path planning (TAP)
    • Assign the track segments to the bits within each rectangular region in track assignment for bits (TAB)
    • Update the track occupancies of all the tracks by the routed \( b \)
  • Outer loop for rip-up and reroute (RR):
    • Add history cost to the edge of BGG
    • Enlarge the frontline size
Our Methodology: MARCH

- Topology-aware Path Planning (TAP):
  - Same layer propagation
  - Layer switching
  - Build routing paths for multi-pin buses
Our Methodology: MARCH

• Topology-aware Path Planning (TAP):
  • Same layer propagation:
    • Propagate the frontline along the track direction ($F_1$ -> $F_2$)
    • Generate a TAP region ($T_1$)
    • Maintain running capacity which will only decrease when some tracks are broken midway
Our Methodology: MARCH

• Topology-aware Path Planning (TAP):
  • Layer switching:
    • Go from one layer to its upper layer or lower layer ($F_2$ on $L_3$ -> $F_3$ on $L_2$) through a switching node
Our Methodology: MARCH

• Topology-aware Path Planning (TAP):
  • Layer switching:
    • Go from one layer to its upper layer or lower layer ($F_2$ on $L_3$ -> $F_3$ on $L_2$)
    • Compute the max number of bits that can pass through the node
      • Without any bottleneck edge: trivial
      • With bottleneck edge(s):
        • With bit order unchanged:
          • Solved by a greedy method
        • With bit order changed:
          • Solved by the right recursive algorithm
        • The efficiency can be guaranteed by pruning
Our Methodology: MARCH

• Topology-aware Path Planning (TAP):
  • Build routing paths for multi-pin buses:
    • Find the path between the source pin and one of the sink pins through the propagation from the source pin
    • Start the propagation from the current path to connect to the next pin
    • Repeat this process until all the pins are connected
Our Methodology: MARCH

• Track Assignment for Bits (TAB)
  • Track segment range estimation:
    • By determining the column/row of G-cells where the bit will be routed
Our Methodology: MARCH

• Track Assignment for Bits (TAB)
  • Track segment range estimation:
    • By determining the column/row of G-cells where the bit will be routed
  • Exact track selection:
    • Three conditions:
      • Satisfy the width constraint
      • Have long enough track segment
      • With as less as possible spacing violations
Our Methodology: MARCH

• Track Assignment for Bits (TAB)
  • Track segment range estimation:
    • By determining the column/row of G-cells where the bit will be routed
  • Exact track selection:
    • Three conditions:
      • Satisfy the width constraint
      • Have long enough track segment
      • With less spacing violations
  • Exact track segment range assignment
    • See the actual routed wires
Our Methodology: MARCH

• Rip-up and Reroute Scheme (RR):
  • Add history cost to the edge of BGG:
    • To eliminate the congested regions on the BGG
    • \( h_{\text{new}} = \alpha \cdot n_{\text{vio}} + \beta \cdot h_{\text{old}} \) where \( n_{\text{vio}} \) is the number of spacing violations on the edge, and \( \alpha \) and \( \beta \) are weights.
  • Enlarge the frontline size:
    • To handle the insufficiency of violation-free routing resources
    • Enlarge the frontline size of the bus by 1 on the layer where:
      • The number of spacing violations in a TAP region is larger than the bit number in 2 successive RR iterations.
Outline

• Problem Formulation

• Our Methodology: MARCH

• Experimental Results

• Conclusion
Experimental Results

<table>
<thead>
<tr>
<th>Characteristics</th>
<th>Metric Weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>bus no.</td>
<td>net no.</td>
</tr>
<tr>
<td>beta1</td>
<td>34</td>
</tr>
<tr>
<td>beta2</td>
<td>26</td>
</tr>
<tr>
<td>beta3</td>
<td>60</td>
</tr>
<tr>
<td>beta4</td>
<td>62</td>
</tr>
<tr>
<td>beta5</td>
<td>6</td>
</tr>
<tr>
<td>final1</td>
<td>18</td>
</tr>
<tr>
<td>final2</td>
<td>70</td>
</tr>
<tr>
<td>final3</td>
<td>47</td>
</tr>
</tbody>
</table>
**Experimental Results**

<table>
<thead>
<tr>
<th></th>
<th>First Place</th>
<th>Second Place</th>
<th>Third Place</th>
<th>MARCH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Croute</td>
<td>Cspace</td>
<td>Cfail</td>
<td>Ctotal</td>
</tr>
<tr>
<td>beta1</td>
<td>689</td>
<td>280</td>
<td>0</td>
<td>969</td>
</tr>
<tr>
<td>beta2</td>
<td>515</td>
<td>760</td>
<td>0</td>
<td>1275</td>
</tr>
<tr>
<td>beta3</td>
<td>1936</td>
<td>0</td>
<td>0</td>
<td>1936</td>
</tr>
<tr>
<td>beta4</td>
<td>2192</td>
<td>0</td>
<td>0</td>
<td>2192</td>
</tr>
<tr>
<td>beta5</td>
<td>119</td>
<td>1848</td>
<td>0</td>
<td>1967</td>
</tr>
<tr>
<td>final1</td>
<td>327</td>
<td>830</td>
<td>2000</td>
<td>3157</td>
</tr>
<tr>
<td>final2</td>
<td>1824</td>
<td>4500</td>
<td>8000</td>
<td>14324</td>
</tr>
<tr>
<td>final3</td>
<td>2966</td>
<td>490</td>
<td>10000</td>
<td>13456</td>
</tr>
<tr>
<td>Avg.</td>
<td>2.130</td>
<td>105.45</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*The scores of top 3 teams of IC/CAD 2018 contest are provided by the contest organizer. A binary is also obtained from the first place to get its runtime information.*
Outline

• Problem Formulation

• Our Methodology: MARCH

• Experimental Results

• Conclusion
Conclusion

• We propose MARCH for bus routing:
  • Have a hierarchical framework (TAP & TAB) for efficiency
  • Route all the bits of a bus concurrently for topology consistency
  • Apply a RR scheme to reduce the routing congestion
  • Performance compared with the top contest teams
    • Reduce spacing violations greatly
    • Avoid any routing failure
    • Have competitive routing costs
    • Have a much shorter runtime
Thank you for attention!
Q & A