Detailed Routing
Detailed Routing

• Find actual geometric layout of each net within assigned routing regions.
• No layouts of two different nets should intersect on the same layer.
• Problem is solved incrementally, one region at a time in a predefined order.
A Routing Example

Channel C
Block
Switchbox
Terminals

Global Routing (a)
Detailed Routing (b)
After Global Routing

• The two-stage routing method is a powerful technique for routing in VLSI circuits.
• During the global routing stage
  – The routing region is partitioned into a collection of rectangular regions.
  – To interconnect each net, a sequence of sub-regions to be used is determined.
  – All nets crossing a given boundary of a routing region are called floating terminals.
  – Once the sub-region is routed, these floating terminals become fixed terminals for subsequent regions.
Order of Routing Regions

- Slicing placement topology
  - Nets can be routed by considering channels 1, 2 and 3 in order.

- Non-slicing placement topology.
- Channels with cyclic constraints.
- Some of the routing regions are to be considered as switchboxes.
Channels and Switchboxes

- There are normally two kinds of rectilinear regions.
  - **Channels**: routing regions having two parallel rows of fixed terminals.
  - **Switchboxes**: generalizations of channels that allow fixed terminals on all four sides of the region.
Routing Considerations

• **Number of terminals**
  – Majority of nets are two-terminal ones.
  – For some nets like clock and power, number of terminals can be very large.
  – Each multi-terminal net can be decomposed into several two-terminal nets.

• **Net width**
  – Power and ground nets have greater width.
  – Signal nets have less width.
Contd.

• **Via restrictions**
  – Regular: only between adjacent layers.
  – Stacked: passing through more than two layers.

• **Boundary type**
  – Regular: straight border of routing region
  – Irregular

• **Number of layers**
  – Modern fabrication technology allows at least five layers of routing.

• **Net types**
  – Critical: power, ground, clock nets
  – Non-critical: signal nets
Routing Models

• **Grid-based model**
  – A grid is super-imposed on the routing region.
  – Wires follow paths along the grid lines.

• **Gridless model**
  – Does not follow the gridded approach.
Models for Multi-Layer Routing

- **Unreserved layer model**
  - Any net segment is allowed to be placed in any layer.

- **Reserved layer model**
  - Certain types of segments are restricted to particular layer(s).
    - Two-layer (HV, VH)
    - Three-layer (VHV, HVH)
Illustration

HVH Model

VHV Model

Unreserved Layer Model
Channel Routing

• **In channel routing, interconnections are made within a rectangular region having no obstructions.**
  – A majority of modern-day ASIC’s use channel routers.
  – Algorithms are efficient and simple.
  – Guarantees 100% completion if channel width is adjustable.

• **Some terminologies:**
  – **Track**: horizontal row available for routing.
  – **Trunk**: horizontal wire segment.
  – **Branch**: vertical wire segment connecting trunks to terminals.
  – **Via**: connection between a branch and a trunk.
Channel Routing Problem :: Terminologies

Net list:: TOP = [1 2 0 2 3 ]
BOT = [3 3 1 1 0 ]
Problem Formulation

- The channel is defined by a rectangular region with two rows of terminals along its top and bottom sides.
  - Each terminal is assigned a number between 0 and N.
  - Terminals having the same label i belong to the same net i.
  - A ‘0’ indicates no connection.
- The netlist is usually represented by two vectors TOP and BOT.
  - TOP(k) and BOT(k) represents the labels on the grid points on the top and bottom sides of the channel in column k, respectively.
Contd.

• **The task of the channel router is to:**
  – Assign horizontal segments of nets to tracks.
  – Assign vertical segments to connect
    • Horizontal segments of the same net in different tracks.
    • The terminals of the net to horizontal segments of the net.

• **Channel height should be minimized.**
• **Horizontal and vertical constraints must not be violated.**
• **Horizontal constraints between two nets:**
  – The horizontal span of two nets overlaps each other.
  – The nets must be assigned to separate tracks.

• **Vertical constraints between two nets:**
  – There exists a column such that the terminal i on top of the column belongs to one net, and the terminal j on bottom of the column belongs to the other net.
  – Net i must be assigned a track above that for net j.
Horizontal Constraint Graph (HCG)

- It is a graph where vertices represent nets, and edges represent horizontal constraints.
Vertical Constraint Graph (VCG)

- It is a directed graph where vertices represent nets, and edges represent vertical constraints.
Two-layer Channel Routing

- **Left-Edge Algorithms (LEA)**
  - Basic Left-Edge Algorithm
  - Left-Edge Algorithm with Vertical Constraints
  - Dogleg Router
- **Constraint-Graph Based Algorithm**
  - Net Merge Channel Router
  - Gridless Channel Router
- **Greedy Channel Router**
- **Hierarchical Channel Router**
Basic Left Edge Algorithm

- **Assumptions:**
  - Only two-terminal nets.
  - No vertical constraints.
  - HV layer model.
  - Doglegs are not allowed.

- **Basic Steps:**
  - Sort the nets according to the x-coordinate of the leftmost terminal of the net.
  - Route the nets one-by-one according to the order.
  - For a net, scan the tracks from top to bottom, and assign it to the first track that can accommodate it.

- **In the absence of vertical constraints, the algorithm produces a minimum-track solution.**
• **Extension to Left-Edge Algorithm**
  – Vertical constraints may exist, but there are no directed cycles in the VCG.
  – Select a net for routing if
    • The x-coordinate of the leftmost terminal is the least.
    • There is no edge incident on the vertex corresponding to that net in the VCG.
  – After routing a net, the corresponding vertex and the incident edges are deleted from the VCG.
  – Other considerations same as the basic left-edge algorithm.
Illustration

VCG


**Dogleg Router**

- **Drawback of LEA**
  - The entire net is on a single track.
  - Sometimes leads to routing with more tracks than necessary.

- Doglegs are used to place parts of the same net on different tracks.

  - A dogleg is a vertical segment that connects two trunks located in two different tracks.
  - May lead to a reduction in channel height.
• Dogleg router allows multi-terminal nets and vertical constraints.
  – Multi-terminal nets can be broken into a series of two-terminal nets.
• Cannot handle cyclic vertical constraints.
Example

No dogleg
3 tracks

With dogleg
2 tracks
Dogleg Router: Algorithm

• **Step 1:**
  – If cycle exists in the VCG, return with failure.

• **Step 2:**
  – Split each multi-terminal net into a sequence of 2-terminal nets.
    • A net \(2 \ldots 2 \ldots 2\) will get broken as \(2a \ldots 2a 2b \ldots 2b\).
  – HCG and VCG gets modified accordingly.

• **Step 3:**
  – Apply the extended left-edge algorithm to the modified problem.
Illustration
Net Merge Channel Router

- Due to Yoshimura and Kuh.
- **Basic idea:**
  - If there is a path of length \( p \) in the VCG, at least \( p \) horizontal tracks are required to route the channel.
  - Try to minimize the longest path in the VCG.
  - Merge nodes of VCG to achieve this goal.
- Does not allow doglegs or cycles in the VCG.
- **How does it work?**
  - Partition the routing channel into a number of regions called “zones”.
  - Nets from adjacent zones are merged.
    - Merged nets are treated as a “composite net” and assigned to a single track.
Key steps of the algorithm:

a) Zone representation
b) Net merging
c) Track assignment

An example:
Step 1: Zone Representation

- Let $S(i)$ denote the set of nets whose horizontal segments intersect column $i$.
- Take only those $S(i)$ which are maximal, that is, not a proper subset of some other $S(j)$.
- Define a zone for each of the maximal sets.
- In terms of HCG / interval graph, a zone corresponds to a maximal clique in the graph.
## Zone Table

<table>
<thead>
<tr>
<th>Column</th>
<th>S(i)</th>
<th>Zone</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{2}</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>{1,2,3}</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>{1,2,3,4,5}</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>{1,2,3,4,5}</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>{1,2,4,5}</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>{2,4,6}</td>
<td>2</td>
</tr>
<tr>
<td>7</td>
<td>{4,6,7}</td>
<td>3</td>
</tr>
<tr>
<td>8</td>
<td>{4,7,8}</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>{4,7,8,9}</td>
<td>4</td>
</tr>
<tr>
<td>10</td>
<td>{7,8,9}</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>{7,9,10}</td>
<td>5</td>
</tr>
<tr>
<td>12</td>
<td>{9,10}</td>
<td></td>
</tr>
</tbody>
</table>

## Zone Representation

```
<table>
<thead>
<tr>
<th>Z1</th>
<th>Z2</th>
<th>Z3</th>
<th>Z4</th>
<th>Z5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

![VCG Diagram]
Step 2: Net Merging

• Let $N_i$ and $N_j$ be two nets for which the following conditions are satisfied:
  – There is no edge between $v_i$ and $v_j$ in HCG.
  – There is no directed path between $v_i$ and $v_j$ in VCG.
• Nets $N_i$ and $N_j$ can then be merged to form a new composite net.
  – Modifies VCG by merging nodes $v_i$ and $v_j$ into a single node $v_{i,j}$.
  – Modifies HCG / zone representation by replacing nodes $v_i$ and $v_j$ by a net $v_{i,j}$, which occupies the consecutive zones including those of nets $N_i$ and $N_j$. 
The process is iterative:
- Pairs of nodes are successively merged.
- At every step of the iteration, in case of multiple choices, merge the net-pair that minimizes the length of the longest path in the VCG.
- That is, the increase in length is minimum.

A result:
- If the original VCG has no cycles, then the updated VCG with merged nodes will not have cycles either.
Contd.

- Iteration 1 of the example:
  - We can merge nets pairs (1,6), (3,6) or (5,6).

Best Choice
• Successive iteration steps:
Step 3: Track Assignment

- Each node in the final graph is assigned a separate track.
- Actually we apply the left-edge algorithm to assign horizontal tracks to the merged nets.
  - The list of nets sorted on their left edges, subject to the vertical constraint, is:
    
    \[ [\ 4-10, \ 1-7, \ 5-6-9, \ 2, \ 3-8 \ ] \]

  Track 1: Nets 4 and 10
  Track 2: Nets 1 and 7
  Track 3: Nets 5, 6 and 9
  Track 4: Net 2
  Track 5: Nets 3 and 8
Greedy Channel Router

• The routing algorithms discussed so far route the channel one net at a time.
  – Based on left-edge algorithm or some of its variation.
• The Greedy Channel Router algorithm routes the channel column by column starting from the left.
  – Apply a sequence of greedy but intelligent heuristic at each column.
  – Objective is to maximize the number of tracks available in the next column.
• Can handle problems with cycles in VCG.
  – May need additional columns at the end of the channel.
Some of the heuristics used:

- Place all segments column by column, starting from the leftmost column.
- Connect any terminal to the trunk segment of the corresponding net.
- Collapse any split net using a vertical segment.
- Try to reduce the distance between two tracks of same net.
- Try to move the nets closer to the boundary which contains the next terminal of that net.
- Add additional tracks if needed.
Channel Routed using a Greedy Router
Hierarchical Channel Router

- **Uses a divide-and-conquer approach.**
  - A routing problem in $m \times n$ grid is reduced to $2 \times n$ grid.
  - Each column in these sub-grids is treated as a *supercell*.
  - Capacity of each vertical boundary is the sum of corresponding boundary capacities.
  - Nets are routed one at a time in the $2 \times n$ grid.
  - Each row of $2 \times n$ is partitioned into $2 \times n$ grid.
  - Terminal position for the new $2 \times n$ grids are defined by the routing in the previous hierarchy.
Example

Reducing \((m \times n)\) grid to \((2 \times n)\) grid
Example (contd.)

First level of hierarchy

Second level of hierarchy
## Comparison of Two-Layer Channel Routers

<table>
<thead>
<tr>
<th></th>
<th>LEA</th>
<th>Dogleg</th>
<th>Net Merge</th>
<th>Greedy</th>
<th>Hierarchical</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
<td>Grid-based</td>
<td>Grid-based</td>
<td>Grid-based</td>
<td>Grid-based</td>
<td>Grid-based</td>
</tr>
<tr>
<td><strong>Dogleg</strong></td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Vertical constraint</strong></td>
<td>No / Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Cyclic constraint</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Three-Layer Channel Routing Algorithms

• Several approaches:
  – Extended Net Merge Channel Router
  – HVH Routing from HV Solution
  – Hybrid HVH-VHV Router
HVH Routing from HV Solution

• Very similar to the Y-K algorithm.
  – Systematically transform a two-layer routing solution into a three-layer routing solution.
  – In Y-K algorithm, nets are merged so that all merged nets forming a composite net are assigned to one track.
  – Here, the composite nets are merged together to form super-composite nets.

• Objective:
  – Reduce the number of super-composite nets.
• Two composite nets in a super-composite net can be assigned to different layers on the same track.

• A track-ordering graph is used to find the optimal pair of composite nets to be merged.
  – Vertices represent the composite (tracks) in a given two-layer solution.
  – The directed edges represent the ordering restrictions on pairs of tracks.
    • Composite interval $t_i$ must be routed above composite interval $t_j$, if there exists a net $N_p \in t_i$ and $N_q \in t_j$, such that $N_p$ and $N_q$ have a vertical constraint.
(a) Track ordering graph

(b) Track ordering graph
An optimal scheduling solution

Graph representation

<table>
<thead>
<tr>
<th>Time</th>
<th>$P_1$</th>
<th>$P_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$t_1$</td>
<td>$t_2$</td>
</tr>
<tr>
<td>2</td>
<td>$t_3$</td>
<td>$t_4$</td>
</tr>
<tr>
<td>3</td>
<td>$t_5$</td>
<td></td>
</tr>
</tbody>
</table>
Limitations of HVH or VHV Router

(a) 

(b) 

(c)
Partitioning for Hybrid Routing
Hybrid HVH-VHV Router

• Uses both HVH and VHV routing schemes.
• Pure HVH and VHV are special cases of Hybrid Router.
• It partitions the channel into two portions – not necessarily of the same size.
  – One portion is for HVH and the other for VHV.
  – One track is required for interconnection between the two portions.
Switchbox Routing

• A switchbox is a generalization of a channel.
  – Has terminals on all four sides.

• More difficult than channel routing problem.
  – Main objective of channel routing is to minimize the channel height.
  – Main objective of switchbox routing is to ensure that all the nets are routed.

• Classification of algorithms:
  – Greedy router
  – Rip up and reroute routers
  – BEAVER (based on computational geometry)
Summary

• The detailed routing problem is solved by routing the channels and switchboxes.
• Routing results may differ based on the routing model used.
  – Grid-based.
  – Based on assigning layer of different net segments.
• The objectives for routing a channel is to minimize channel density, the length of routing nets, and the number of via’s.
• The main objective of channel routing is to minimize the total routing area.
• The objective of switchbox routing is to determine the routability.