Global Routing

Readings: Chapter 5

Determine overall path of all routes
Pick channels to route through

Seeks to reduce delay, channel width

Partitioning
Floorplanning
Placement
Global Routing
Detailed Routing
Compaction

Metal Rules

First-level metal
minimum width: 3λ.
minimum spacing: 3λ.
metal1 size for contact: 6λ.
metal1 size for via: 7λ.

Second-level metal
minimum width 4λ.
minimum spacing 3λ.
metal2 size for via: 7λ.
**Grid Abstraction for Routing**

Abstract routing regions as grid, with spacing set to ensure design rules met.

**Maze Router**

Breadth-first search along "wavefront"
Maze Router Algorithm

\[ Index = 1; \]
Mark source as \textit{Index};

While destination unmarked {
    mark all unmarked neighbors of cells marked with \textit{Index} as \textit{Index}+1;
    \textit{Index}++;
}

Trace monotonically decreasing path from destination back to source;

Unmark all nodes;

This algorithm is optimal for 2-terminal nets.

Maze Router Performance Concerns

Must store information on all grid locations encountered

For a length L path, must consider \(O(L^2)\) locations
Maze Router Acceleration: Bidirectional Search

Search from both source and destination until wavefronts meet

Maze Routing Acceleration: A*/Detour Numbers

Cost of each location includes minimum distance to destination
Line Search Algorithms - Removing the Grid

Are all possible grid locations important/interesting?

Track Graphs

Extend vertical & horizontal lines from source, target & obstacle corners
Greedy Multi-terminal Routing

Route until first terminal found.

Unmark all nodes. Nodes on previous path marked as 1. Continue routing.

Sub-optimality of Greedy Multi-terminal Routing

Multi-terminal routing is greedy, and route to first location may not be best for subsequent destinations.
Minimum Spanning Tree

While net routing not completed
  Find shortest distance from any connected to any not connected node.
  Add corresponding route

Algorithm optimal if we disallow non-terminal split points
Algorithm is quick

Steiner Trees

Allow split points ("Steiner Points") anywhere on the chip
MSTs vs. Minimum Steiner Trees

Efficiency:
- MSTs found in $O(E \log_2 E)$
- Finding Minimum Steiner Trees are NP-Hard

Quality:
- MSTs are never shorter than Minimum Steiner Trees
  - A MST is a Steiner Tree with 0 Steiner nodes
  - MSTs are at worst 50% longer than Minimum Steiner Trees

Idea: approximate Steiner Tree with MST, then look for helpful Steiner Nodes

Approximating Steiner Trees from MSTs

Build Track Graph
Find MST of source & destinations
while benefit found {
  Foreach unused Steiner point
  find MST of current tree + Steiner point
  select Steiner point providing best benefit
}

Note: Min. Steiner tree always has points at intersections on Track Graph
Approximating Steiner Trees from MSTs (cont.)

Won’t always find optimal results:

Approximate:

Exact: