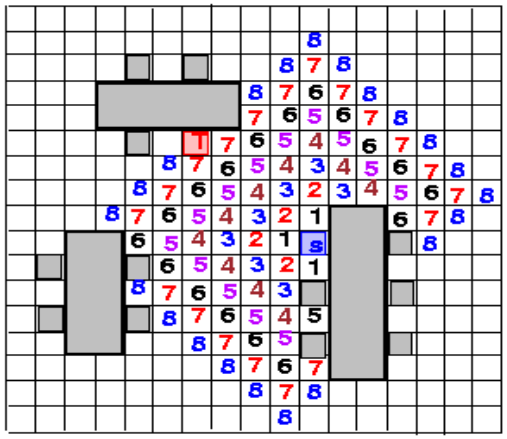
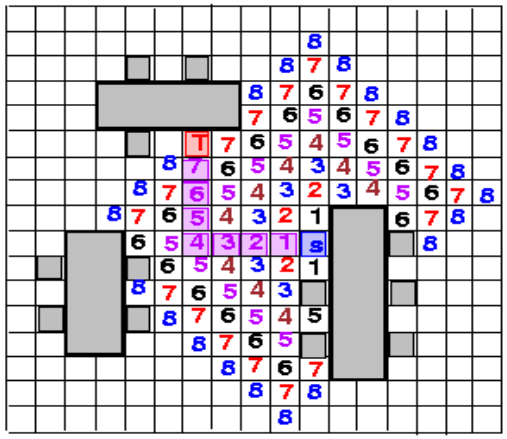
Unit 5D: Maze (Area) and Global Routing

- Course contents
  - Routing basics, maze (area) routing, line searching/probing
  - Global routing
  - Detailed routing
    - But (main stream - track assignment + rip-up-reroute)
- Readings
  - Chapters 9.1, 9.2, 9.5



Filling – wave expansion



Retrace

Multiple solutions in tracing back. Which one to choose? Less bends?

How about multiple layers?

Any concerns about this algorithm?

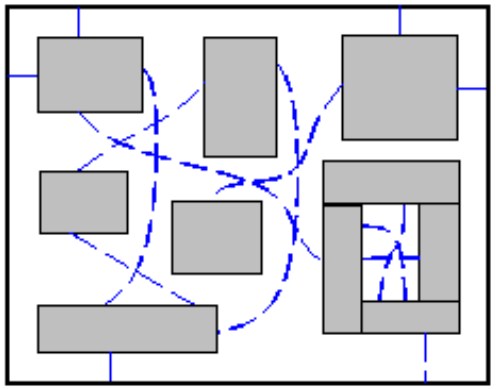
Routing

How to model routing regions? Channels?
Non-slicing? Can we route over the macros?

placement

- Generates a "loose" route for each net.
- Assigns a list of routing regions to each net without specifying the actual layout of wires.

global routing



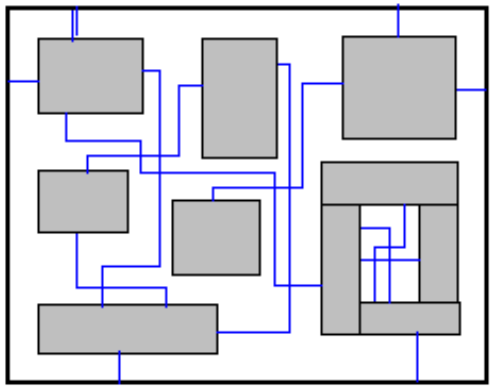
Global routing

How about layer assignments?

detailed routing

- Finds the actual geometric layout of each net within the assigned routing regions.

How to create wires without DRC errors?



Detailed routing

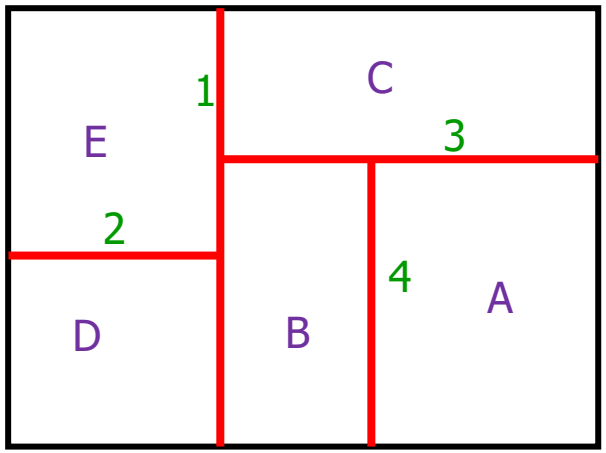
compaction

Routing & Layout

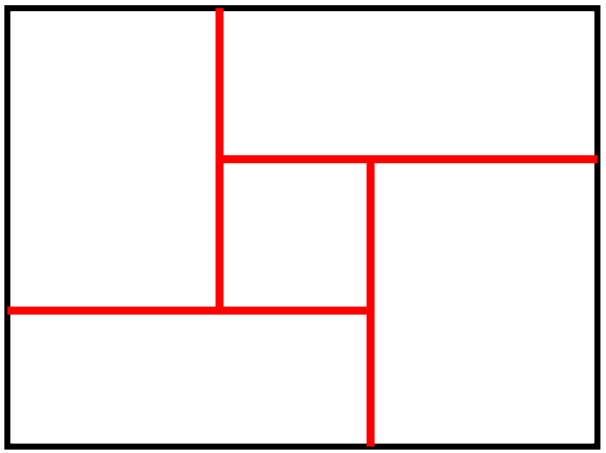
- Chip layout size can be changed (variable sized)
  - But which channel can be expanded? And how to handle incremental routing?
  - In the past, designers will follow slicing structure, whose channels can be adjusted. However, not all the chips can follow slicing structure.

- Chip layout size is fixed – nowadays it is the majority
  - Two styles:
    - Every net is connected; then, do rip-up-reroute to fix DRC violations
    - Route as many nets as possible; if violations, leave nets open

Slicing



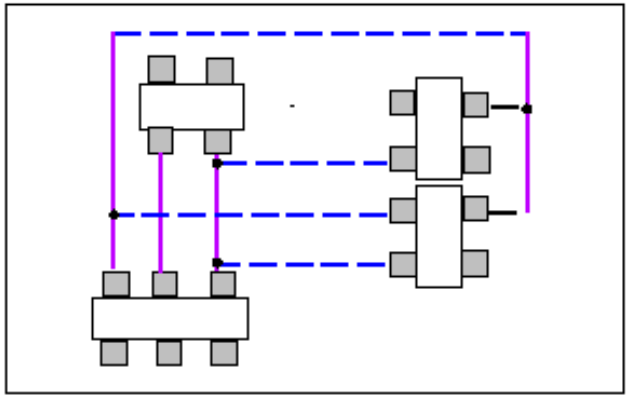
Non sliciable



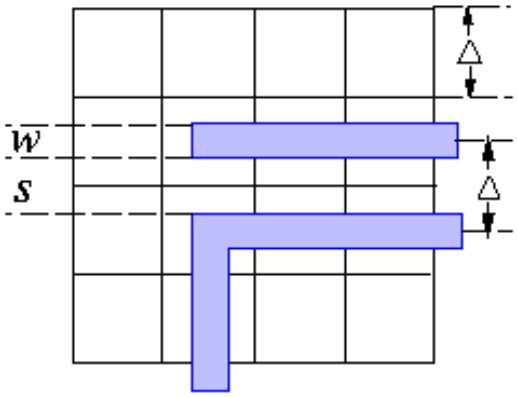
If the channel-4 is routed, then area (A, B) can be packed. Then, we route channel-3, area (C, (A+B)) can be decided. Then, we do channel-2 to get area (D, E). Lastly we route channel-1

Routing Constraints

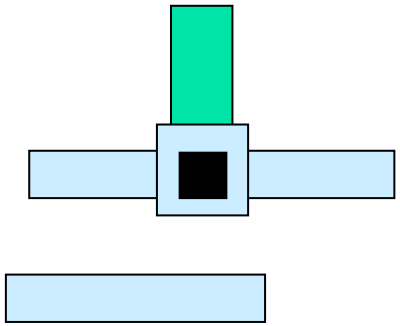
- 100% routing completion with clean DRC + area minimization(?), under a set of constraints:
  - Placement constraint: usually based on fixed placement
  - Number of routing layers; routing blockages;
  - Geometrical constraints: must satisfy design rules
  - Timing constraints (performance-driven routing): must satisfy delay constraints; clock tree routing requires special considerations
    - Crosstalk?
    - Process variations?



Two–layer routing



Geometrical constraint

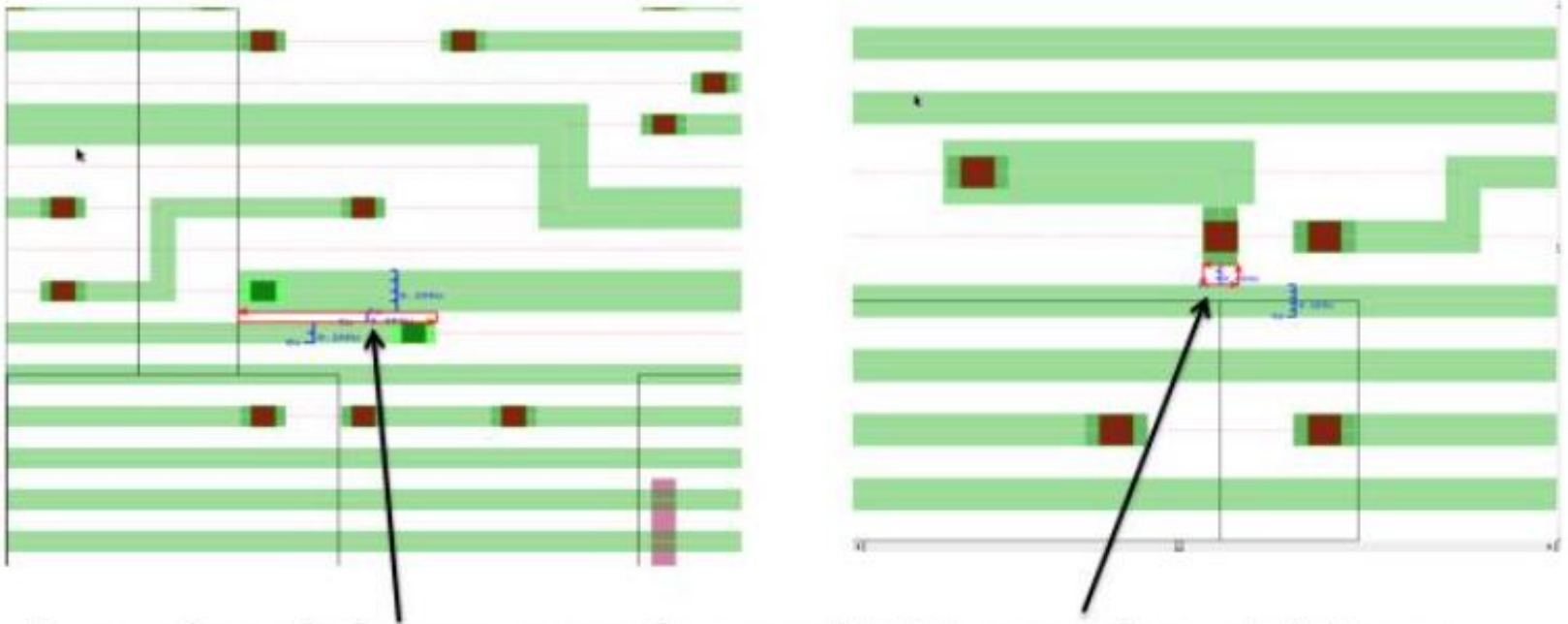


Contact will impact pitches

Digress To Talk About DRC

- Important topic for router development

Min Spacing and End-Of-Line Spacing Violation Examples



Example **minimum spacing** and **EOL spacing** violations between routing objects in congested areas. Many such violations are in the vicinity of pins assigned an NDR rule.

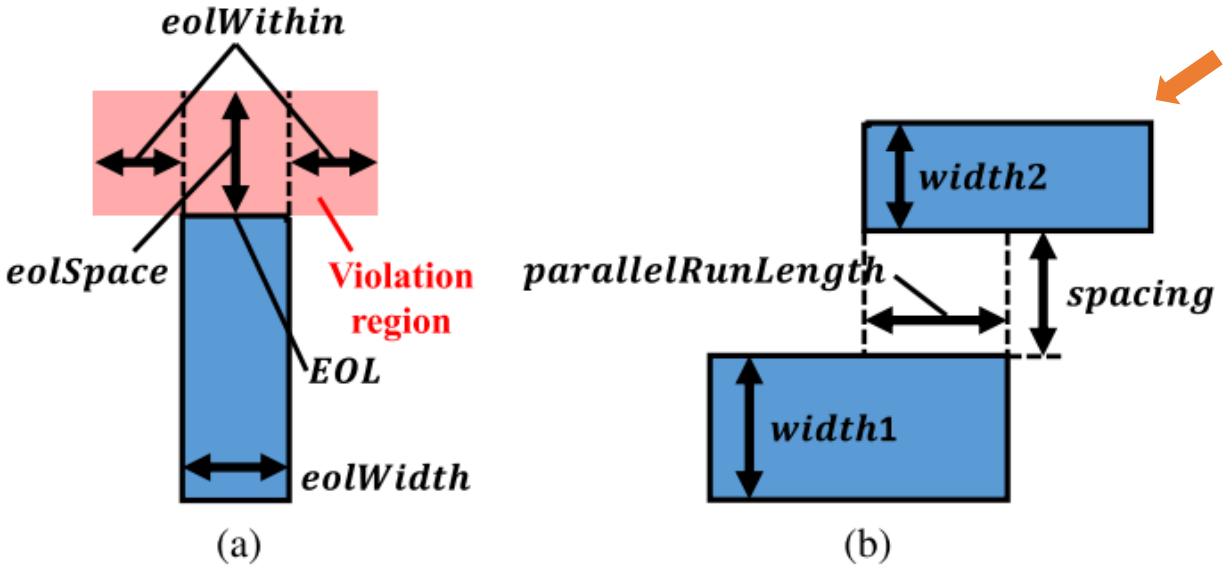


Fig. 2. Example of (a) EOL spacing and (b) parallel-run spacing.

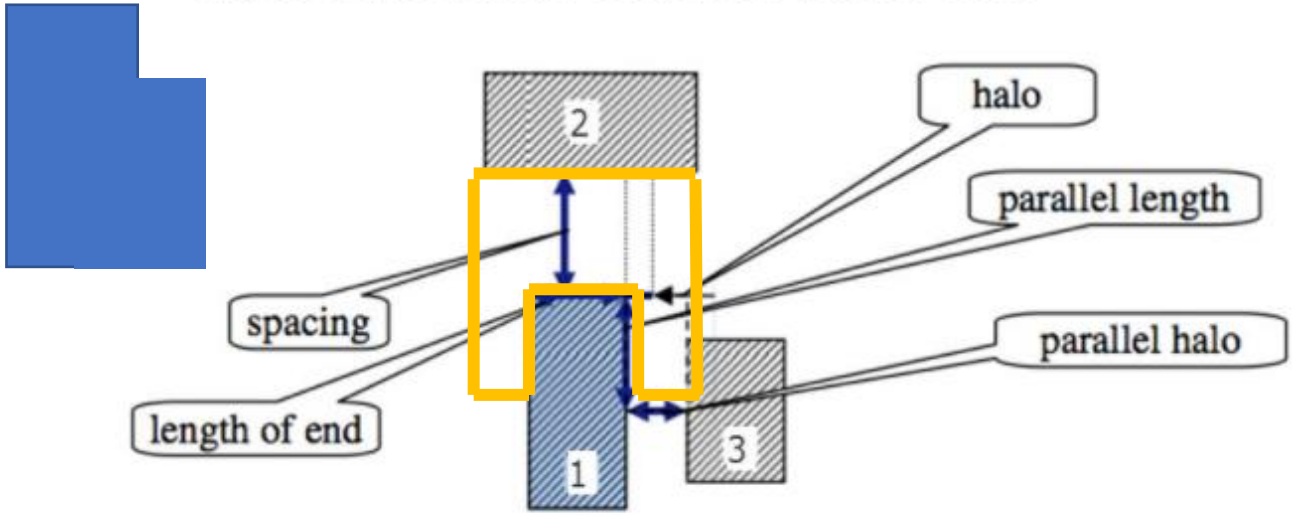
2) EOL Spacing: A metal end is an EOL if its width is shorter than eolWidth. EOL is required to preserve a spacing greater than or equal to eolSpace beyond the EOL anywhere less than the eolWithin distance, as Fig. 2(a) shows

Parallel-Run Spacing: For two metal objects with parallelRunLength (i.e., the projection length between them), there is a spacing requirement, as Fig. 2(b) shows. The value of parallel-run spacing rule depends on the widths of the two metal rectangles.

Rule Examples (2) – An Important Rule

End of line rule

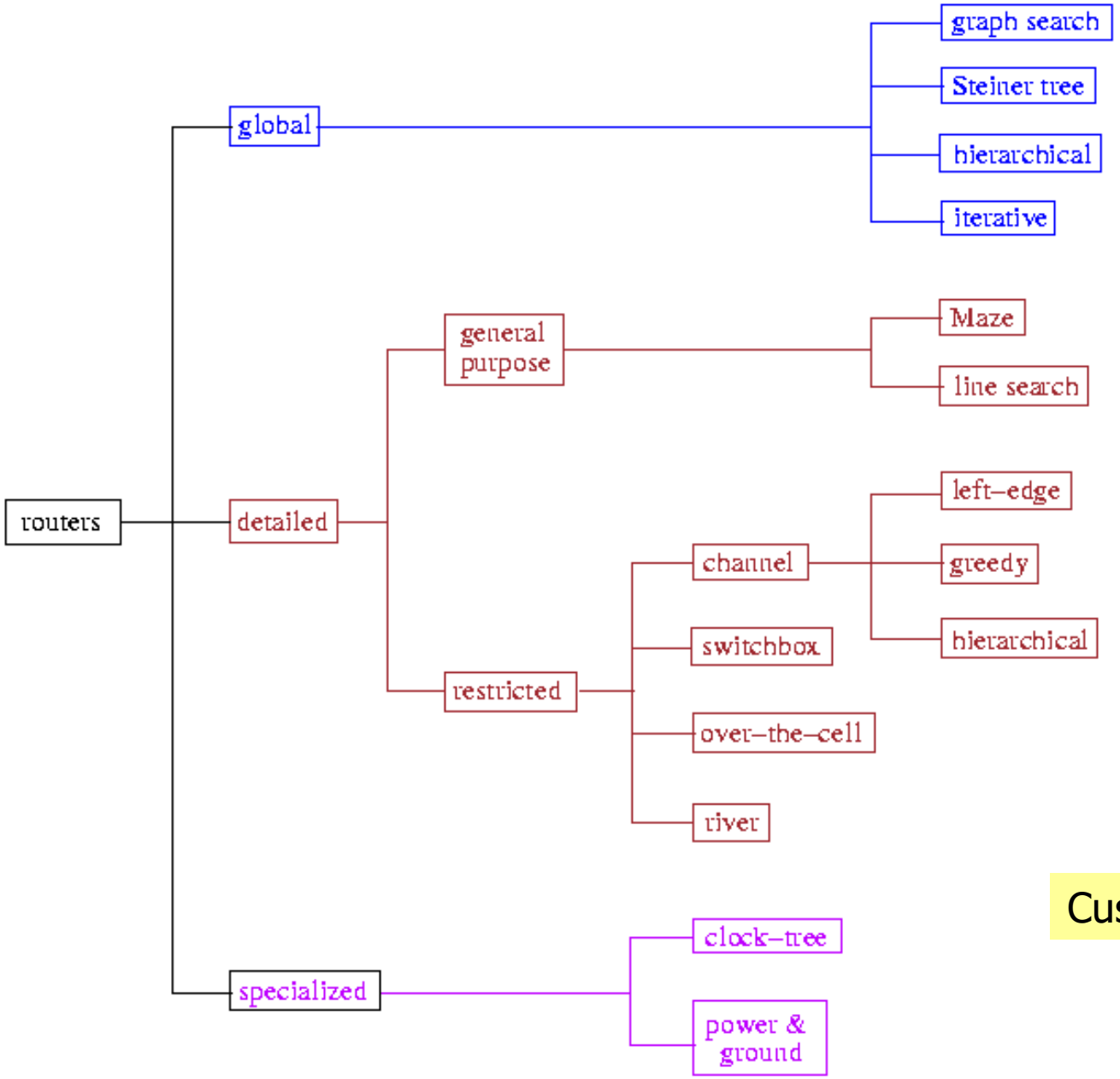
- EOL spacing applied to objects 1 and 2:
  - — As object 3 overlaps the parallel length from the top of edge 1, EOL spacing between objects 1 and 2 will be required.
  - — Object 3 must remain outside the parallel halo.



How are we going to check it?

We can guess which foundries were better on one tech node thru their DRM rules

Classification of Routing



Track assignment, Rip-up-reroute

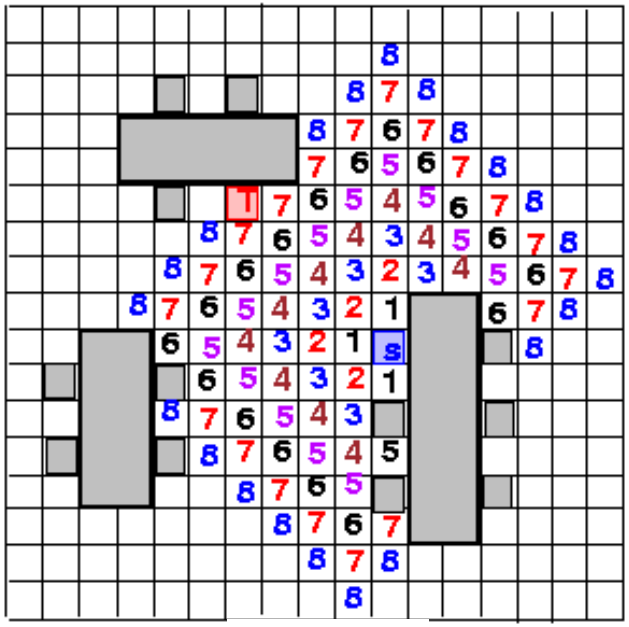
Custom routing

Maze Router: Lee Algorithm

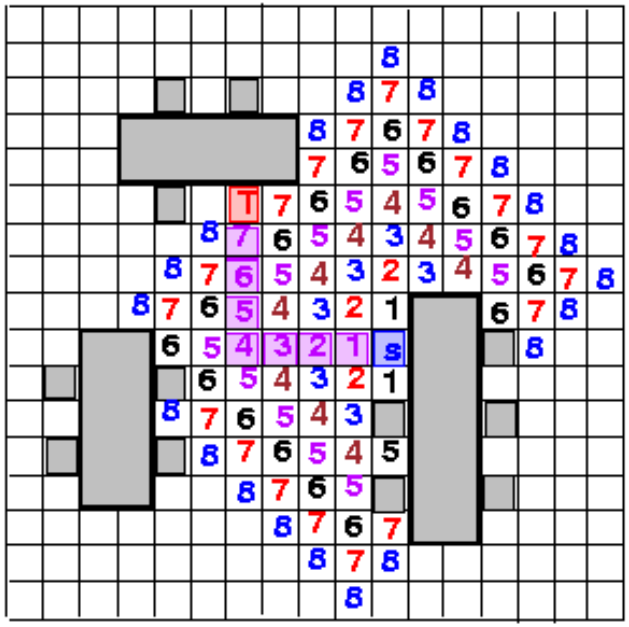
- Lee, “An algorithm for path connection and its application,” *IRE Trans. Electronic Computer*, EC-10, 1961.
- Discussion mainly on single-layer routing
- **Strengths**
  - Guarantee to find connection between 2 terminals if it exists.
  - Guarantee minimum path.
- **Weaknesses**
  - Requires large memory for dense layout
  - Slow
- Applications: path finding, global routing, detailed routing
- Need enhancements to handle multi-layers routing

Lee Algorithm

- Find a path from *S* to *T* by “wave propagation”.



Filling



Retrace

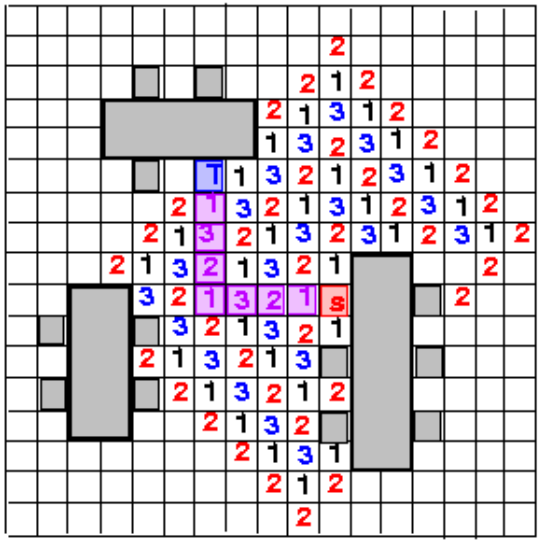
- Time & space complexity for an $M \times N$ grid: $O(MN)$ (**huge!**)
- If considering multiple layers, then even more expensive

Reducing Memory Requirement

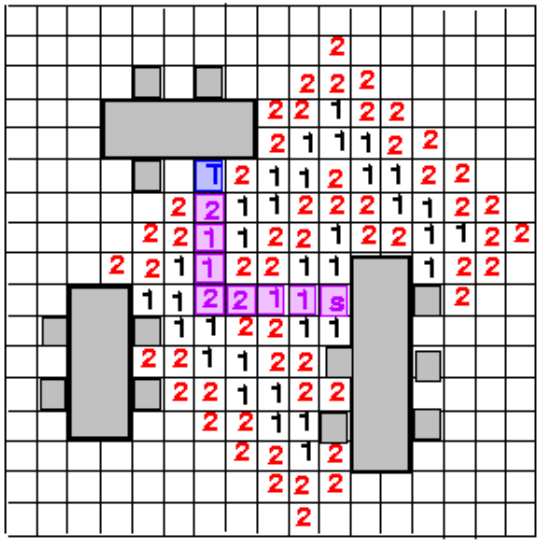
- Akers's Observations (1967)

  Instead of 32/64-bits

  - Adjacent labels for *k* are either *k*-1 or *k*+1.
  - Want a labeling scheme such that each label has its preceding label different from its succeeding label.
- Way 1: coding sequence 1, 2, 3, 1, 2, 3, …; states: 1, 2, 3, *empty*, *blocked* (3 bits required)
- Way 2: coding sequence 1, 1, 2, 2, 1, 1, 2, 2, …; states: 1, 2, *empty*, *blocked* (need only 2 bits)



Sequence: 1, 2, 3, 1, 2, 3, ...

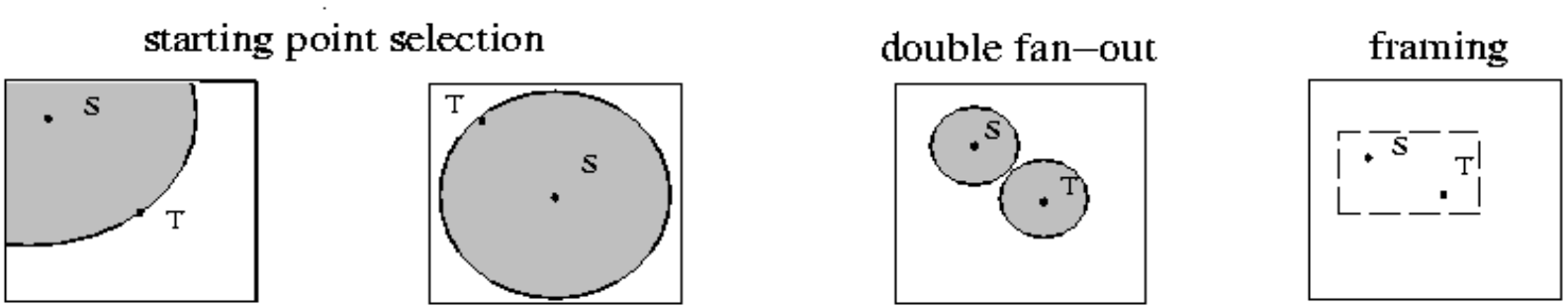


Sequence: 1, 1, 2, 2, 1, 1, 2, 2, ...

Reducing Running Time

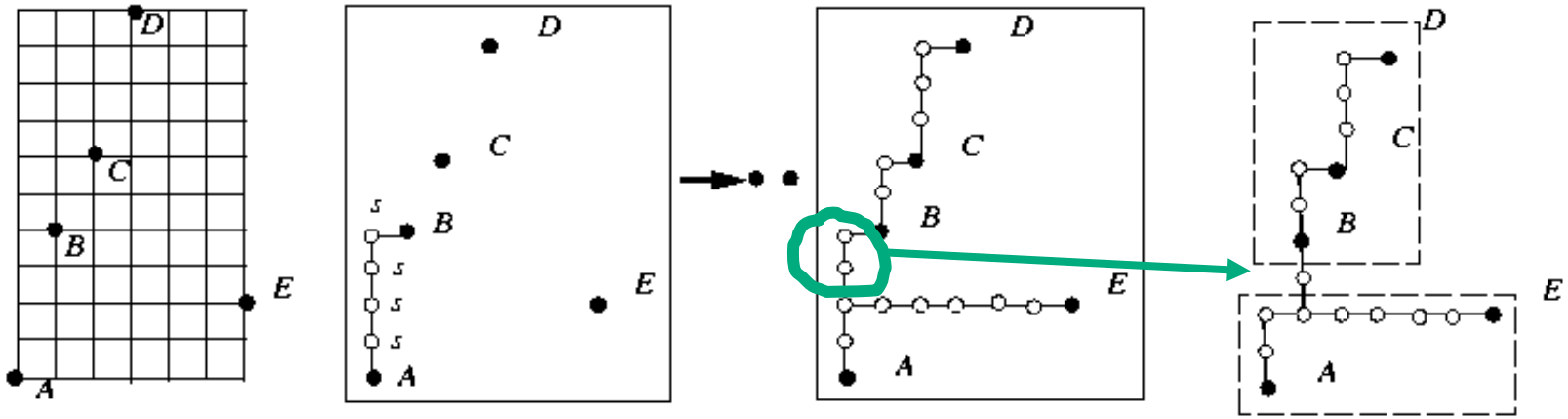
A* algorithm

- Starting point selection: Choose the (waveform?) point farthest from the center of the grid as the starting point.
- Double fan-out: Propagate waves from both the source and the target cells.
- Framing: Search inside a rectangle area 10--20% larger than the bounding box containing the source and target.
  - Need to enlarge the rectangle and redo if the search fails.



Connecting Multi-Terminal Nets

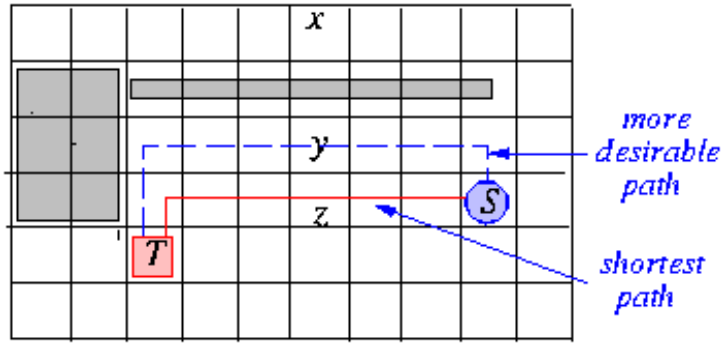
- Step 1: Propagate wave from the source *s* to the closet target.
- Step 2: Mark ALL cells on the path as *s*.
- Step 3: Propagate wave from ALL *s* cells to the other cells.
- Step 4: Continue until all cells are reached.
- Step 5: Apply heuristics to further reduce the tree cost.

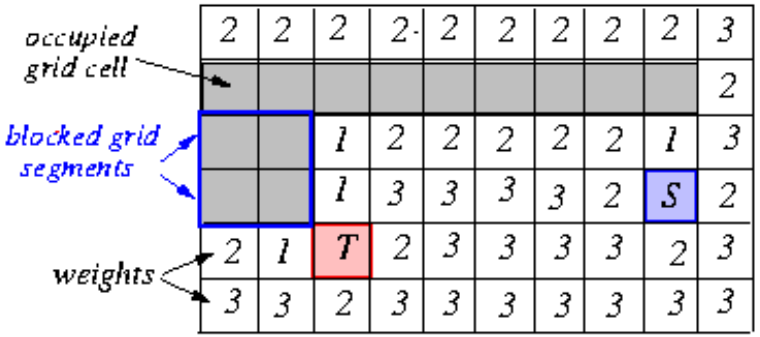


When to "beautify" routing patterns?

Routing on a Weighted Grid

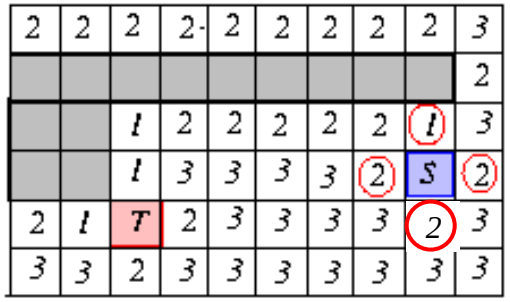
- Motivation: finding more desirable paths
- *weight*(*grid cell*) = *# of unblocked grid cell segments* - 1



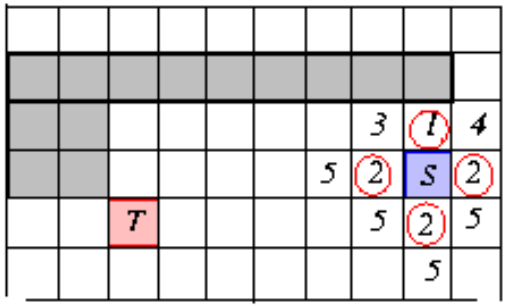


Initialize cell weights

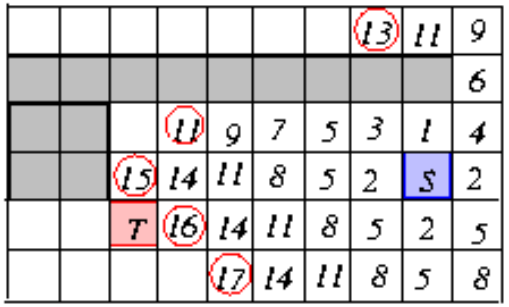
A Routing Example on a Weighted Grid

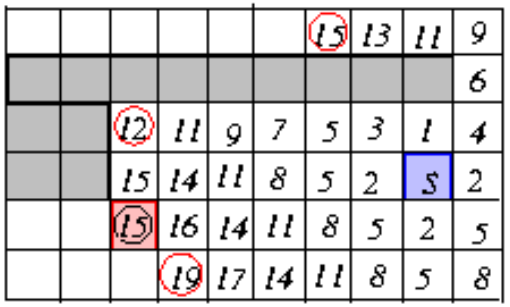


initialize cell weights

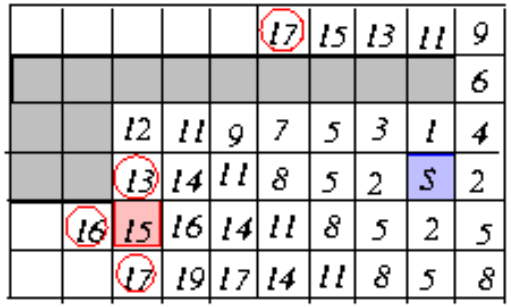


wave propagation

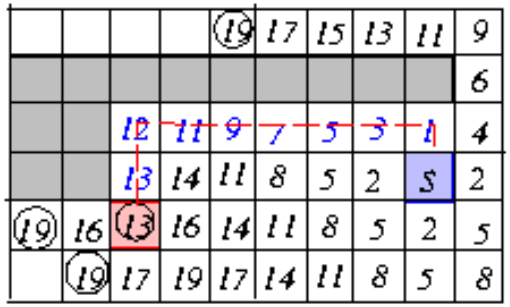




first wave reaches the target



finding other paths



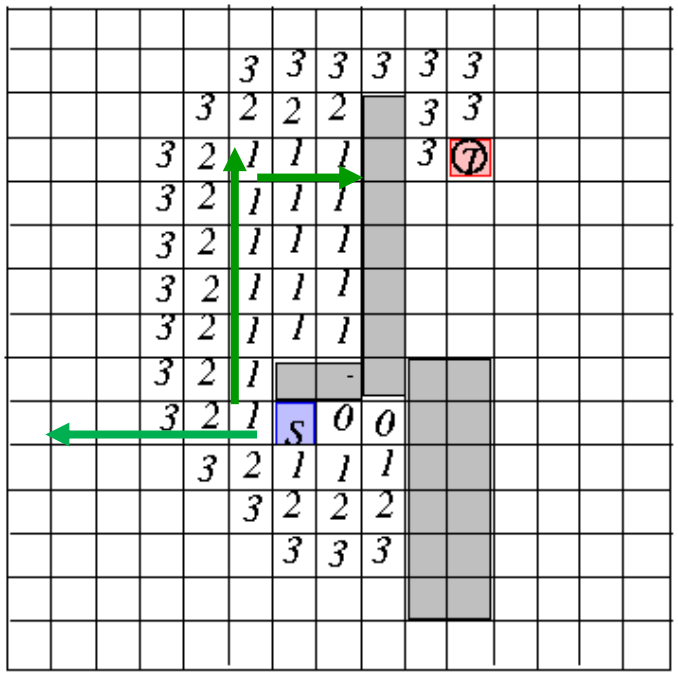
min−cost path found

Hadlock's Algorithm

- Hadlock, “A shortest path algorithm for grid graphs,” *Networks*, 1977
- Uses detour number (instead of labeling wavefront in Lee's router)
  - Detour number, $d(P)$: # of grid cells directed **away from** its target on path P.
  - $MD(S, T)$: the Manhattan distance between S and T.
  - Path length of P, $l(P)$: $l(P) = MD(S, T) + 2\,d(P)$.
  - $MD(S, T)$ fixed! $\Rightarrow$ Minimize $d(P)$ to find the shortest path.
  - For any cell labeled i, label its adjacent unblocked cells **away from** T i+1; label i otherwise.
- Time and space complexities: $O(MN)$, but substantially reduces the # of searched cells.
- Finds the shortest path between S and T.

Hadlock's Algorithm (cont'd)

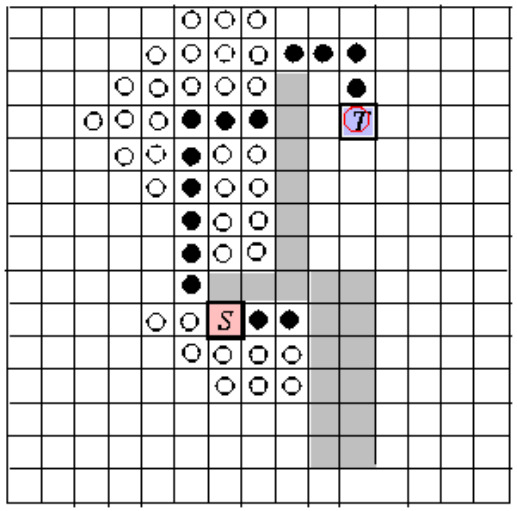
- $d(P)$: # of grid cells directed **away from** its target on path P.
- $MD(S, T)$: the Manhattan distance between S and T.
- Path length of P, $l(P)$: $l(P) = MD(S, T) + 2d(P)$.
- $MD(S, T)$ fixed! $\Rightarrow$ Minimize $d(P)$ to find the shortest path.
- For any cell labeled i, label its adjacent unblocked cells **away from** T i+1; label i otherwise.

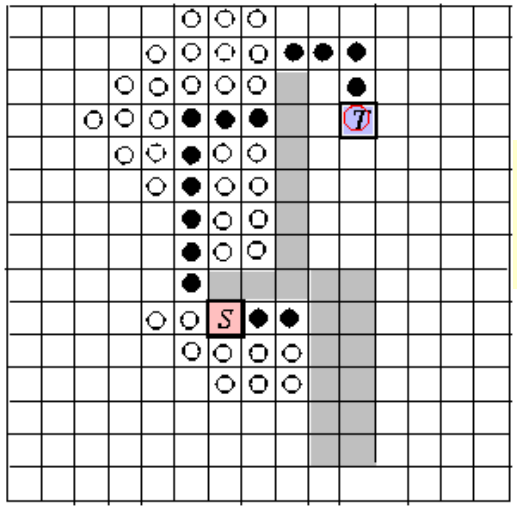


Kind of line search

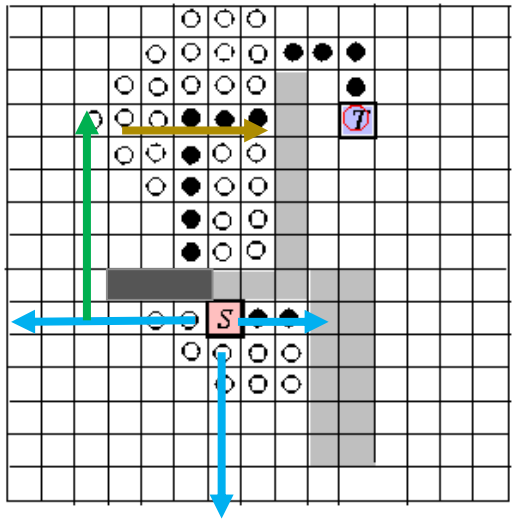
Soukup's Algorithm

- Soukup, “Fast maze router,” DAC-78.
- Combined breadth-first and depth-first search.
  - Depth-first (**line**) search is first directed toward target *T* until an obstacle or *T* is reached.
  - Breadth-first (Lee-type) search is used to “bubble” around an obstacle if an obstacle is reached.
- Time and space complexities: *O*(*MN*), but 10--50 times faster than Lee's algorithm.
- Find **a** path between *S* and *T*, but may not be the shortest!



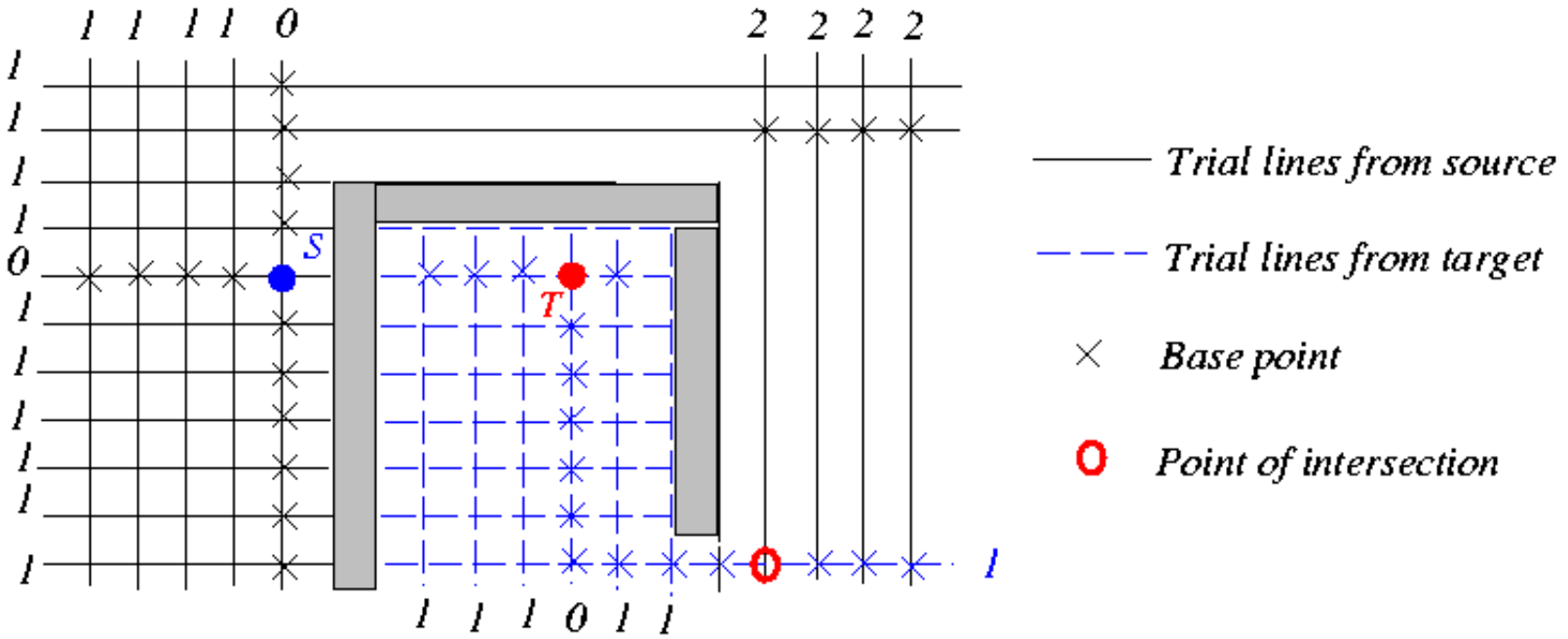


If blockage is bigger



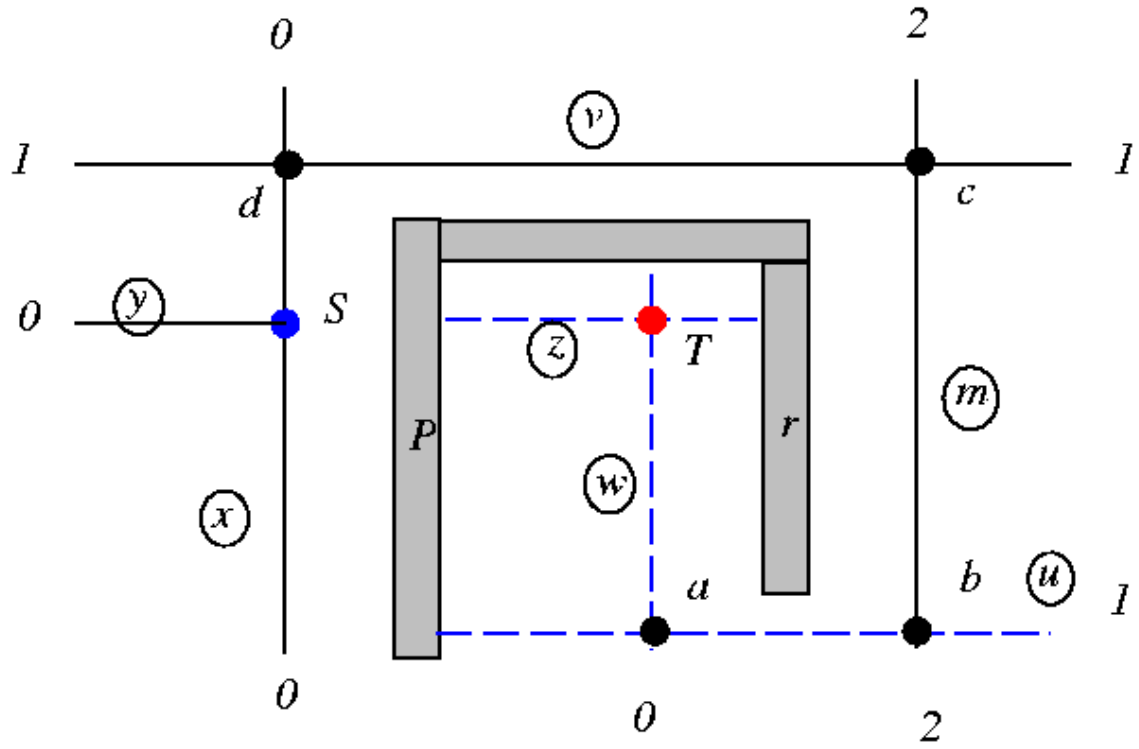
Mikami-Tabuchi's Algorithm

- Mikami & Tabuchi, “A computer program for optimal routing of printed circuit connectors,” *IFIP*, H47, 1968.
- Every grid point is an escape point.



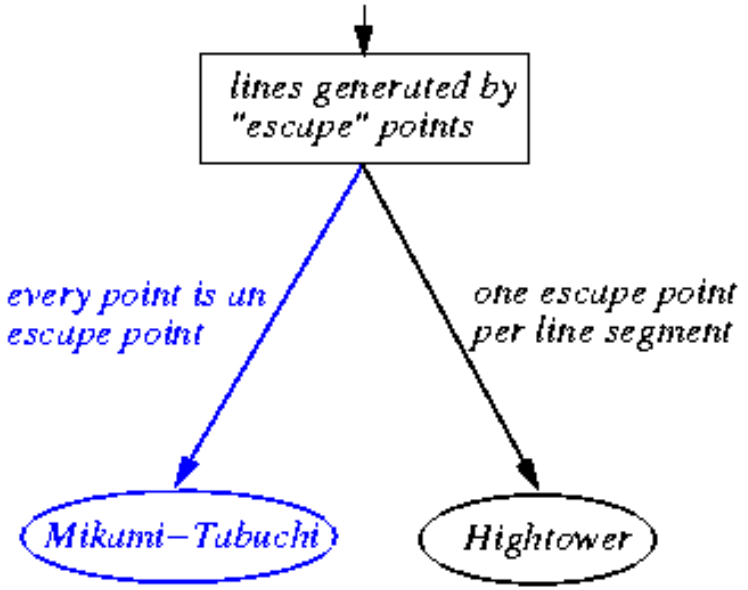
Hightower's Algorithm

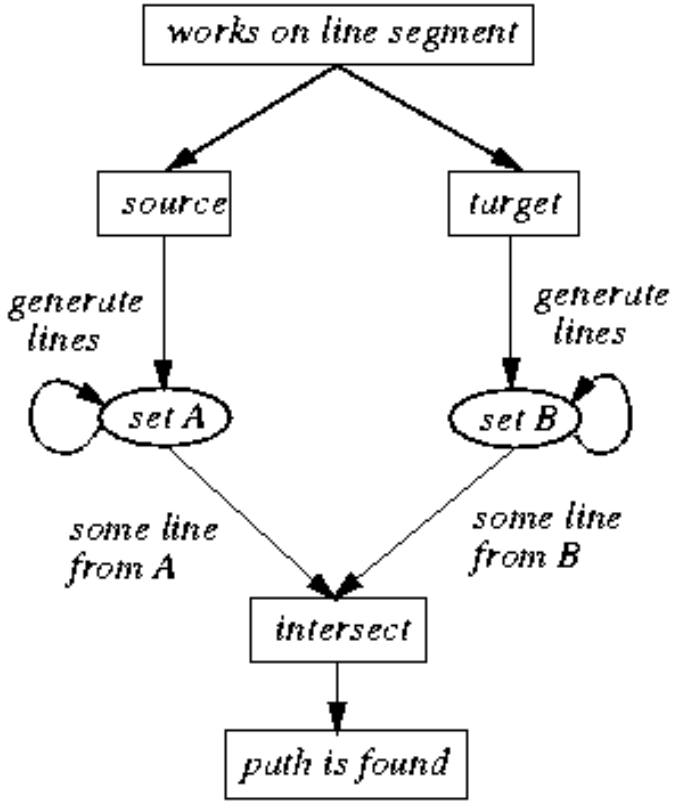
- Hightower, “A solution to line-routing problem on the continuous plane,” DAC-69.
- A single escape point on each line segment.
- If a line parallels to the blocked cells, the escape point is placed just past the endpoint of the segment.



Routing order will affect the result. First net’s result becomes new blockages for 2nd net

Features of Line-Search Algorithms





- Time and space complexities: $O(L)$, where L is the # of line segments generated.
- Also called line probing; later on area probing ⇔ Used in **custom routing** a lot

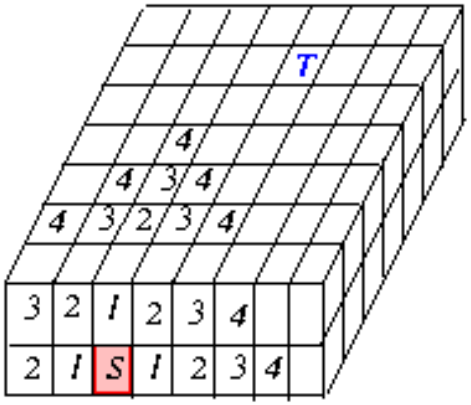
Comparison of Algorithms

	Maze routing			Line search	
	Lee	Soukup	Hadlock	Mikami	Hightower
Time	$O(MN)$	$O(MN)$	$O(MN)$	$O(L)$	$O(L)$
Space	$O(MN)$	$O(MN)$	$O(MN)$	$O(L)$	$O(L)$
Finds path if one exists?	yes	yes	yes	yes	no
Is the path shortest?	yes	no	yes	no	no
Works on grids or lines?	grid	grid	grid	line	line

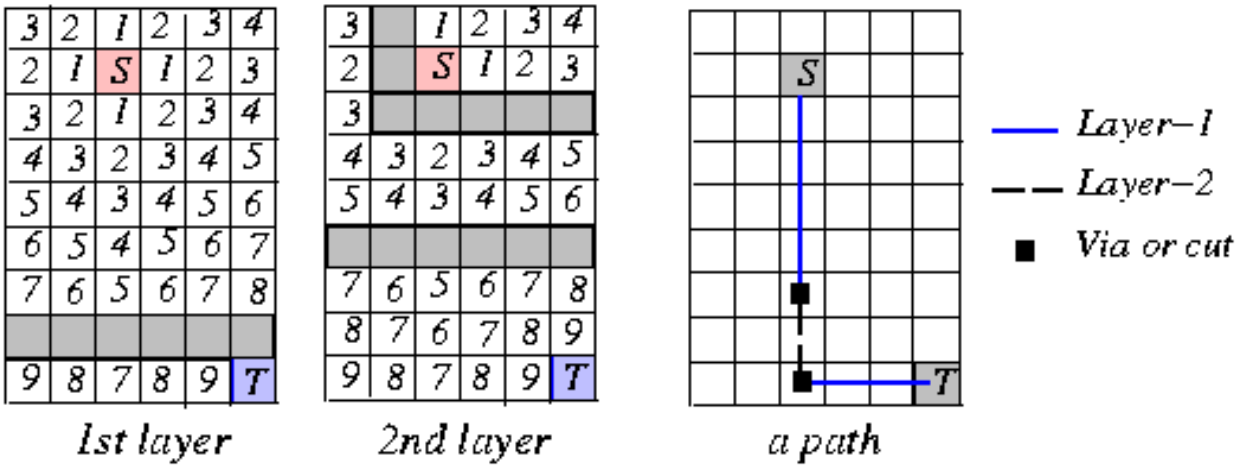
- Soukup, Mikami, and Hightower all adopt some sort of line-search operations $\Rightarrow$ cannot guarantee shortest paths.

Multi-layer Routing

- 3-D grid:

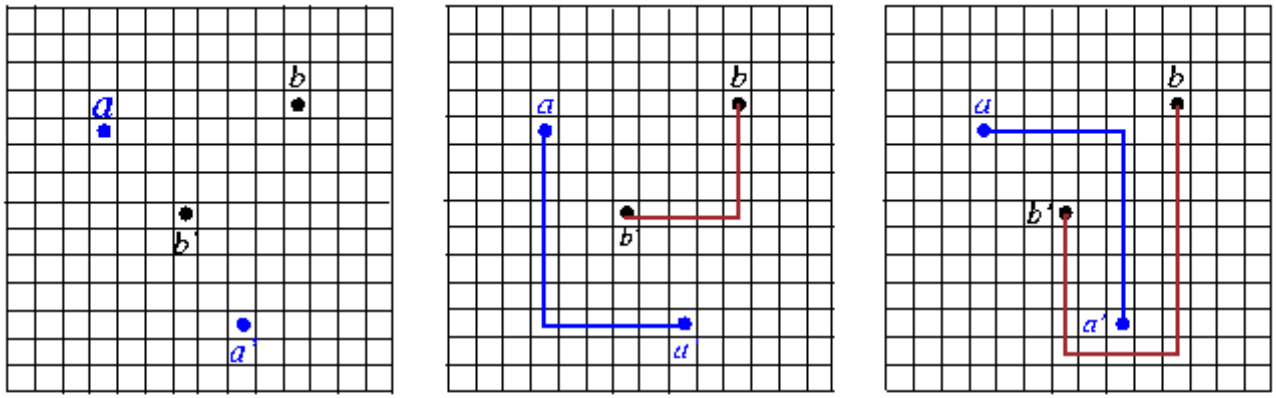


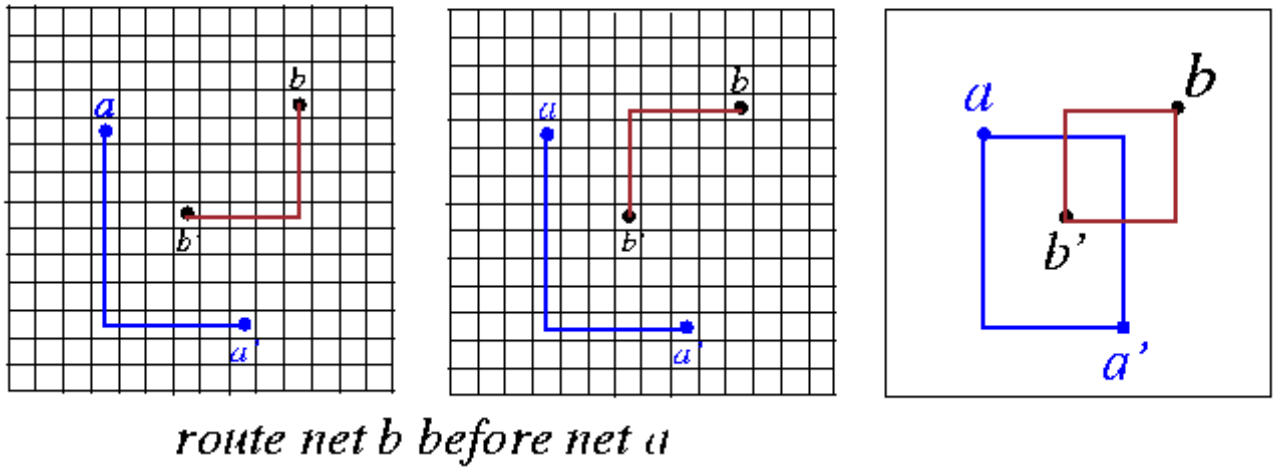
- Two planar arrays:
  - Neglect the weight for inter-layer connection through via
  - Pins are accessible from both layers



Net Ordering

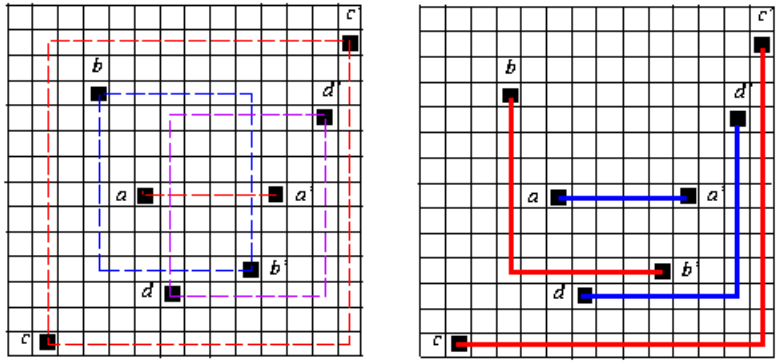
- Net ordering greatly affects routing solutions.
- In the example, we should route net *b* before net *a*. Why?





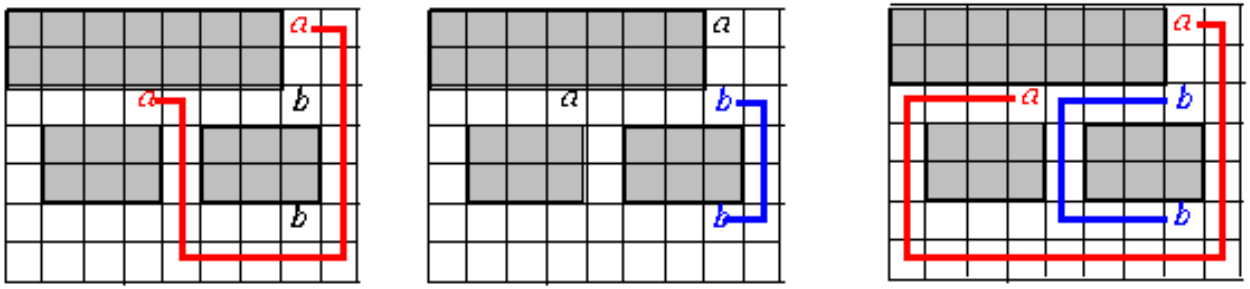
Net Ordering (cont'd)

- Order the nets in the ascending order of the # of pins within their bounding boxes. (nets with more pins will need to wait.)
- Order the nets in the ascending (or descending??) order of their lengths. (shorter one goes 1st)
- Order the nets based on their timing criticality.



routing ordering: a (0) –> b (1) –> d (2) –> c (6)

- A mutually intervening case:



a prevents routing of b *b prevents routing of a* *a feasible routing*

Routing

How to model routing regions? Channels?
Non-slicing? Can we route over the macros?

placement

- Generates a "loose" route for each net.
- Assigns a list of routing regions to each net without specifying the actual layout of wires.

global routing

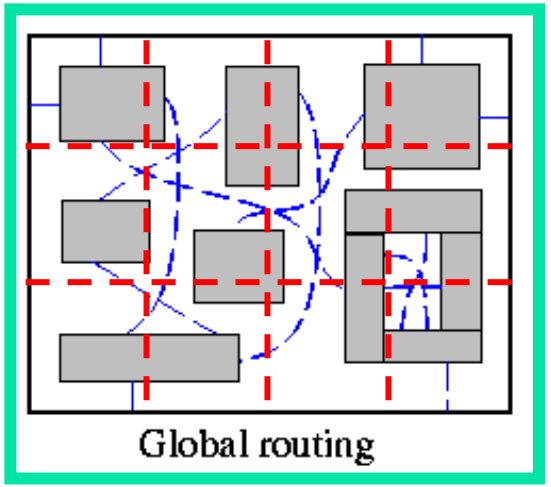
How about layer assignments?

detailed routing

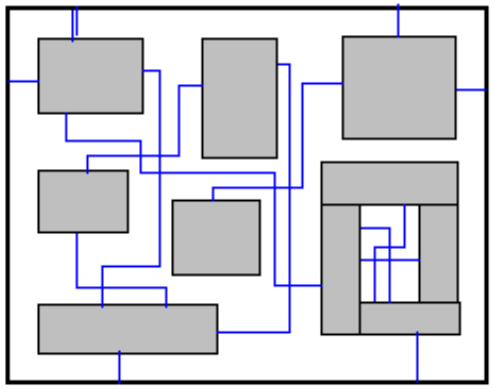
- Finds the actual geometric layout of each net within the assigned routing regions.

compaction

How to create wires without DRC errors?

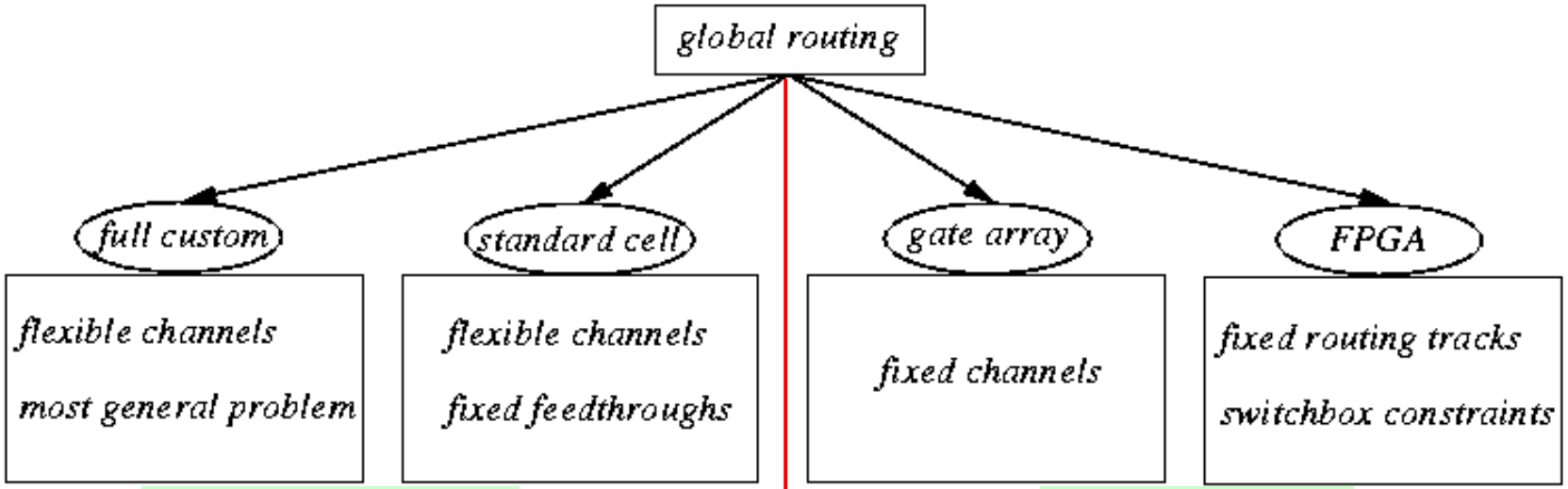


Global routing



Detailed routing

Global Routing in different Design Styles

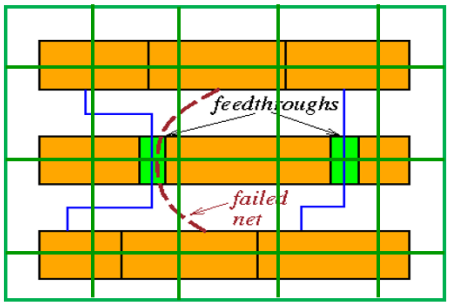


Variable size

Fixed size

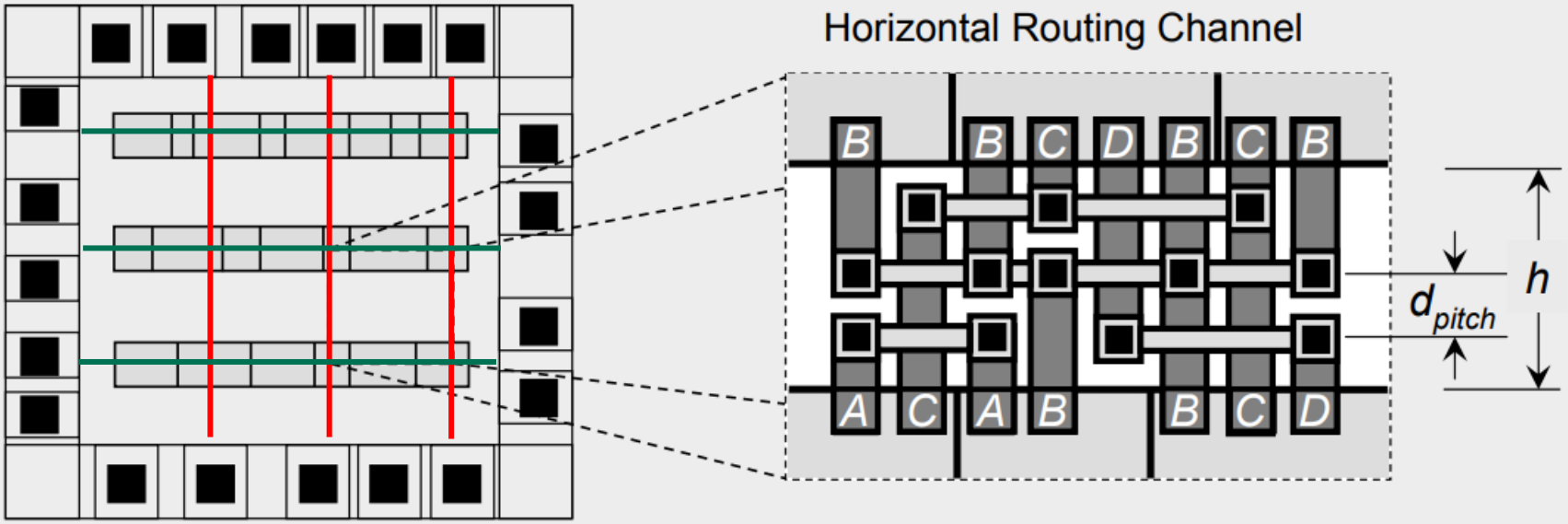
Fixed size

Die-to-die wiring (packaging)



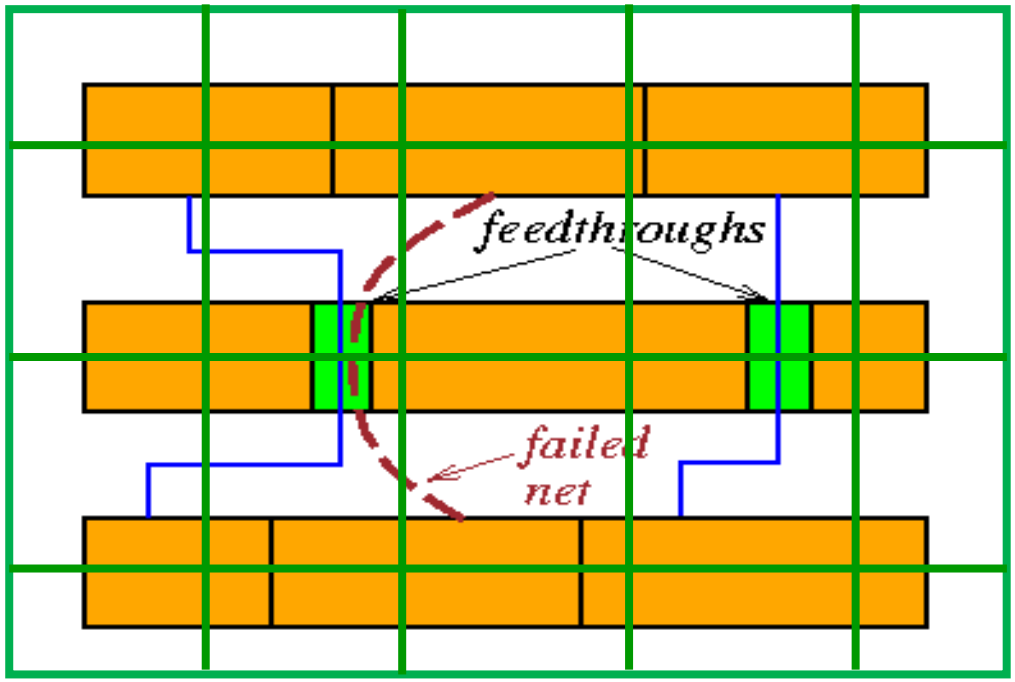
Capacity

Number of available routing tracks or columns



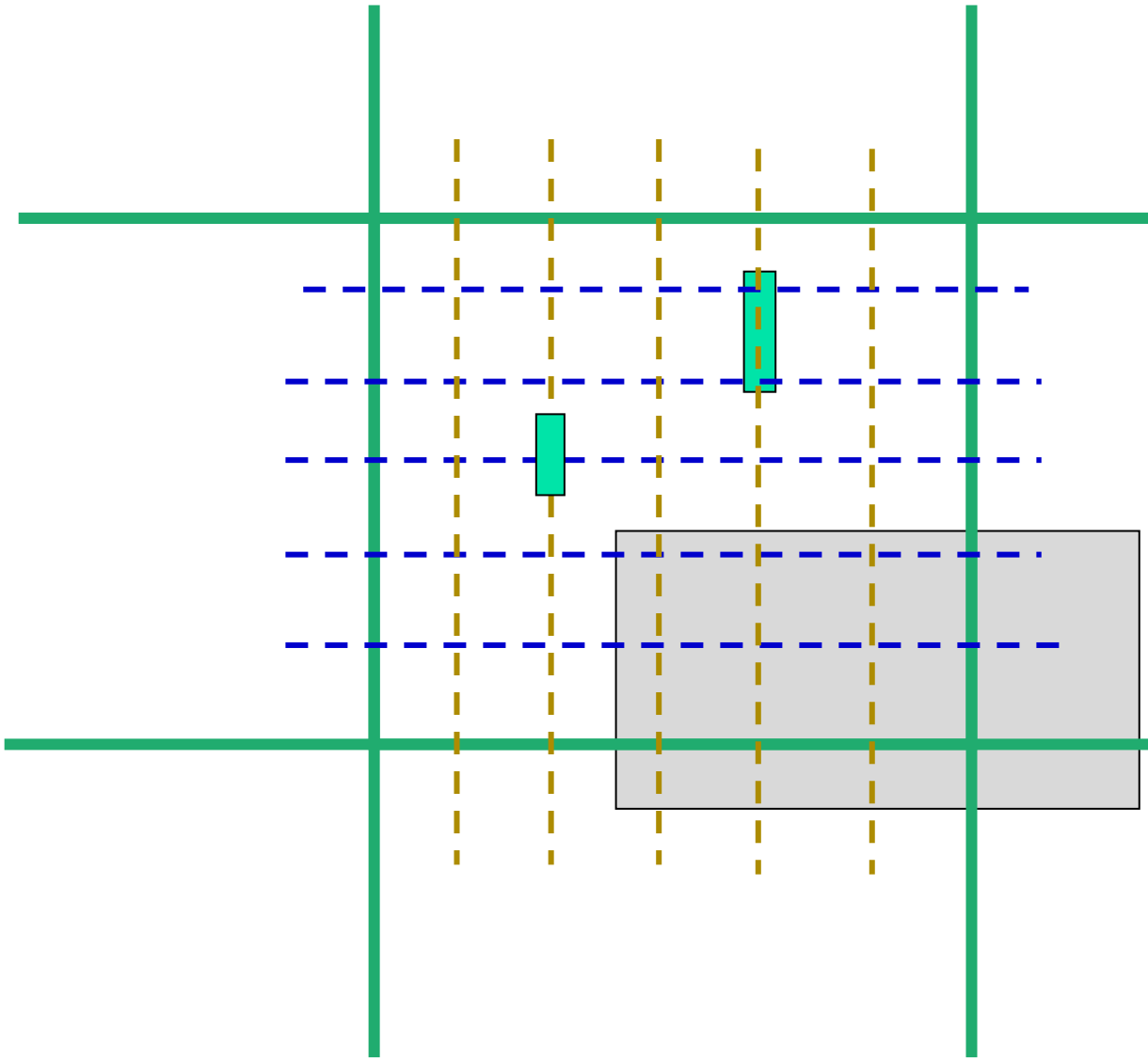
Global Routing in Standard Cell

- Objective
  - Minimize total channel height.
  - Assignment of **feedthrough**: Placement? Global routing?
- For high performance,
  - Minimize the maximum wire length.
  - Minimize the maximum path length.



A possible way of GRcells

How To Model In A GCell



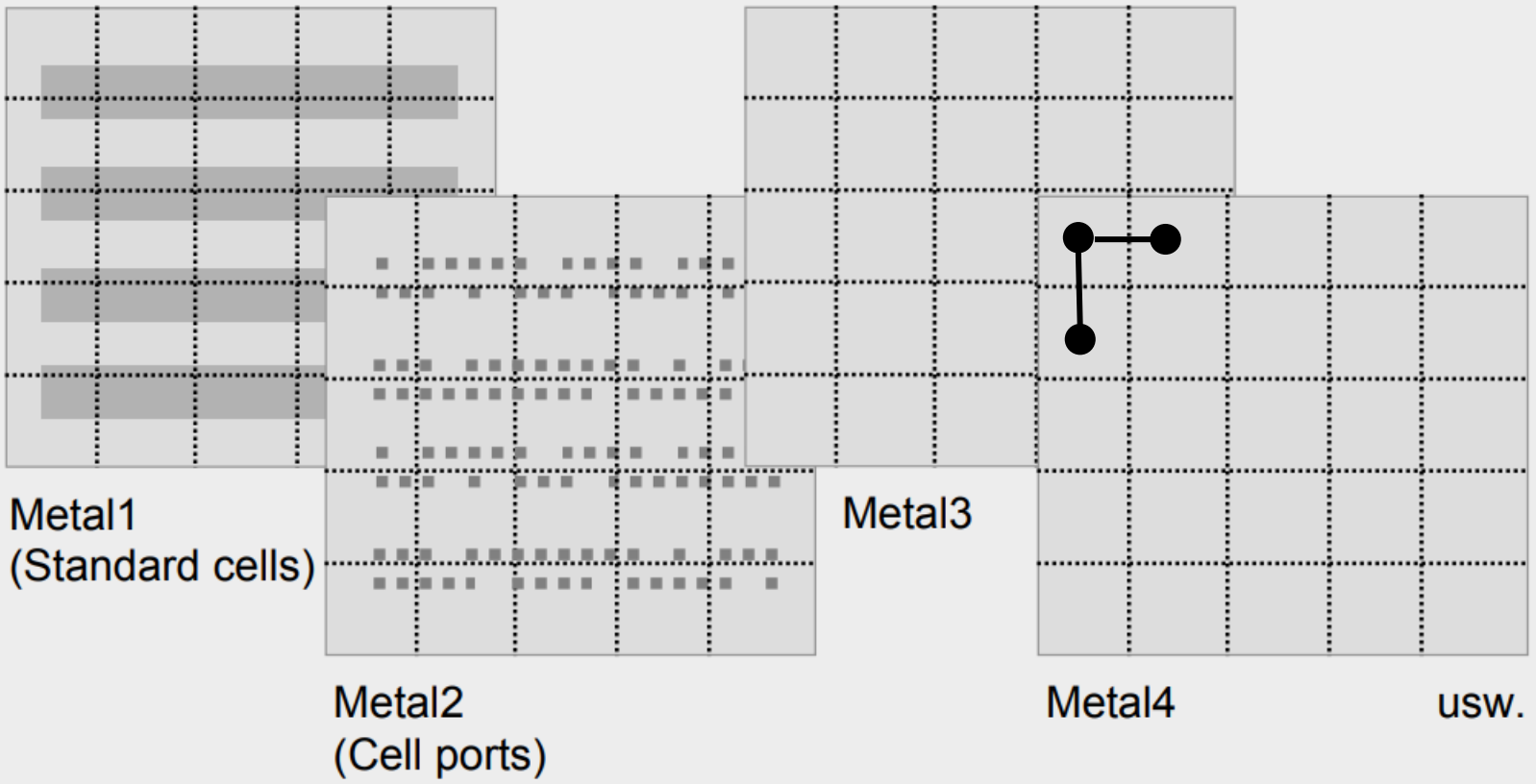
Available tracks
@ pins
How big the blockages

Horizontal track capacities can be different on left & right sides (full, partial)

# of pins in a global cell will impact the results

vias in a global cell will also impact the results

Gcells (Tiles) with standard cells

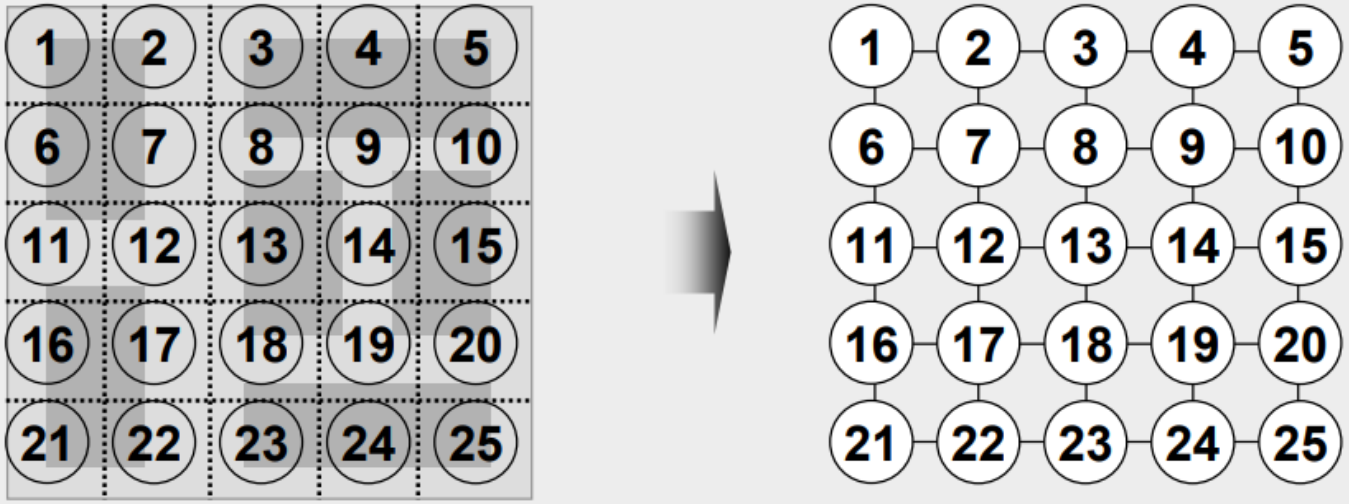


Most Of The SOC Chips

- Consist of many standard cells and macros
- Use the global routing cells as shown in previous slide

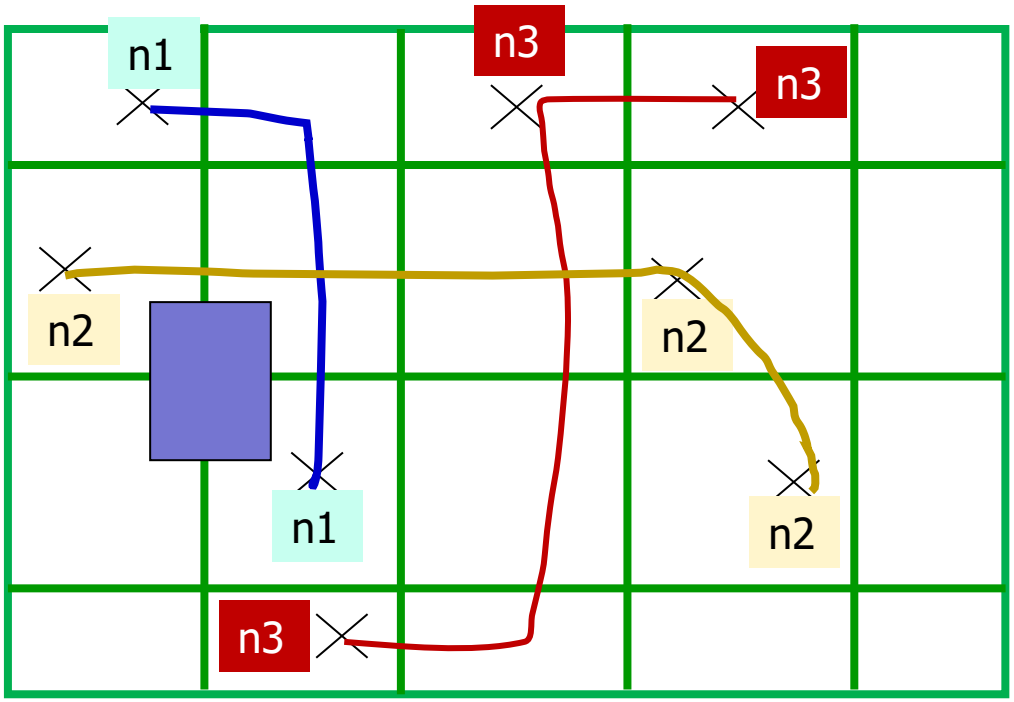
Grid graph model

Modeling the Gcells needs tuning



ggrid = (*V*,*E*), where the nodes $v \in V$ represent the routing grid cells (*gcells*) and the edges represent connections of grid cell pairs (v_i, v_j)

(My Preference)
Pictorial Explanation Of Global Routing



How is a net to be connected?

- Global route is to find loose routing for all the nets such that
  - Every net is connected and
  - no boundary capacities are exceeded
- Can we smooth out the boundary congestions?
  - Important
- Considering the timing or criticality of the nets
- GR uses lots of memory (more than that for DR)

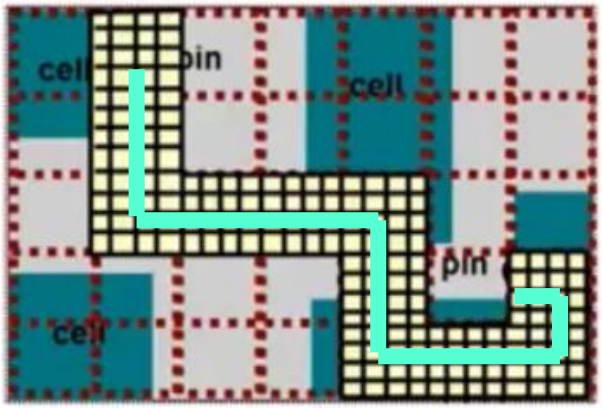
Global-Routing Problem

- Given a netlist N={N_1, N_2, …, N_n }, a routing graph G = (V, E), find a Steiner tree T_i for each net N_i, $1 \le i \le n$, such that $U(e_j) \le c(e_j)$, $\forall\ e_j \in E$ and $\sum_{i=1}^{n} L(T_i)$ is minimized, where
  - $c(e_j)$: capacity of edge e_j;
  - x_{ij}=1 if e_j is in T_i; x_{ij} = 0 otherwise;
  - $U(e_j) = \sum_{i=1}^{n} x_{ij}$: # of wires that pass through the channel corresponding to edge e_j; (demand)
  - $L(T_i)$: total wirelength of Steiner tree T_i.

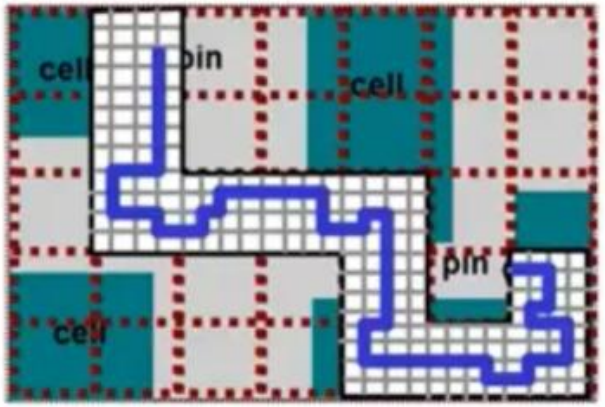
- For high-performance, the maximum wirelength ($\max_{i=1}^{n} L(T_i)$) is minimized (or the longest path between two points in T_i is minimized).

An Algorithm At The High Level

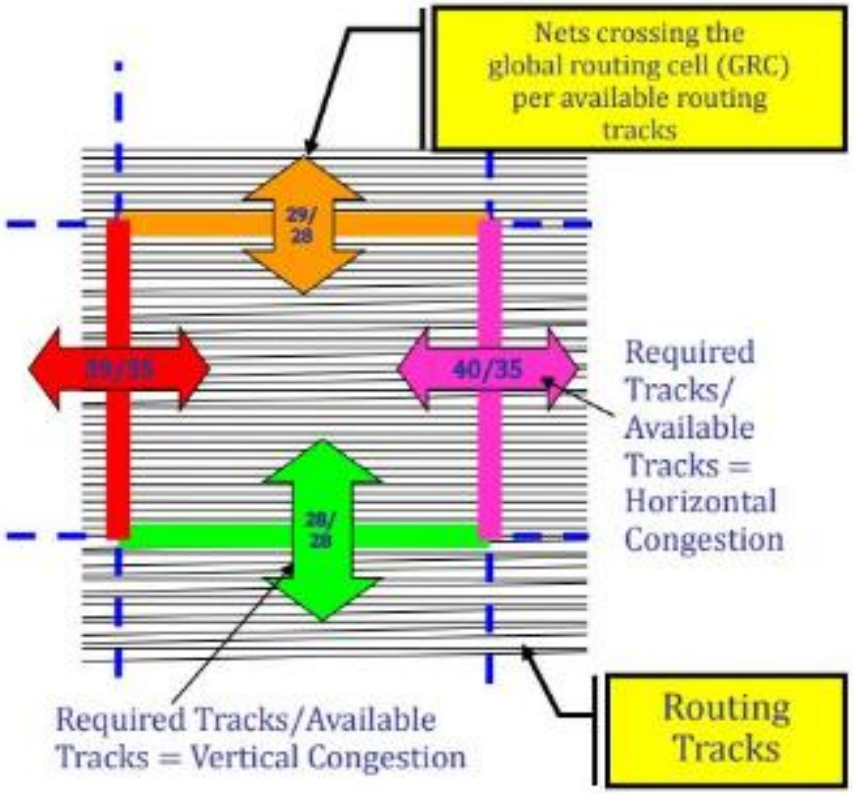
- Deciding the global routing grids – can be non-uniform
  - How to decide the Gcell sizes
- Finding the track capacity at the Gcell boundaries
  - Model the wire crossings, vias, pins around the Gcell centers
- Critical nets global are routed first – to get shorter length
  - Can use maze search, rectilinear Steiner/spanning tree, …
- Route other signal nets
  - If going thru congested areas, higher cost
- Reserving some slacks at Gcell boundaries
- After all the nets are global routed, find out any boundary overflows
- Decide how to rip-up-reroute to even out the congestions
- Better to consider layer assignments
- Sometimes need to go with hierarchical approach



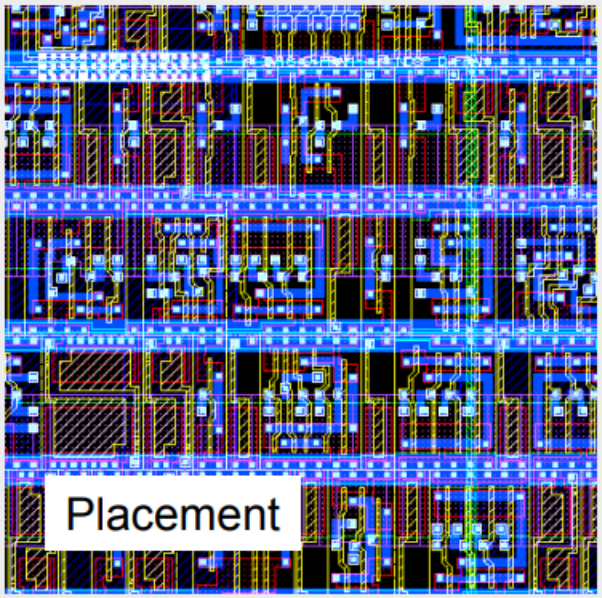
Global router tells us to search for detailed paths only here

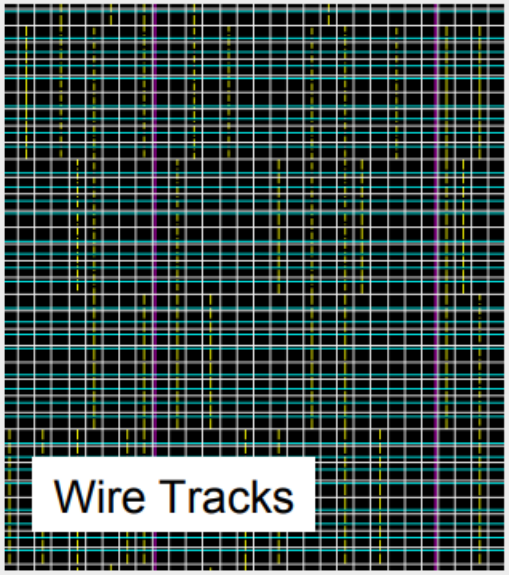


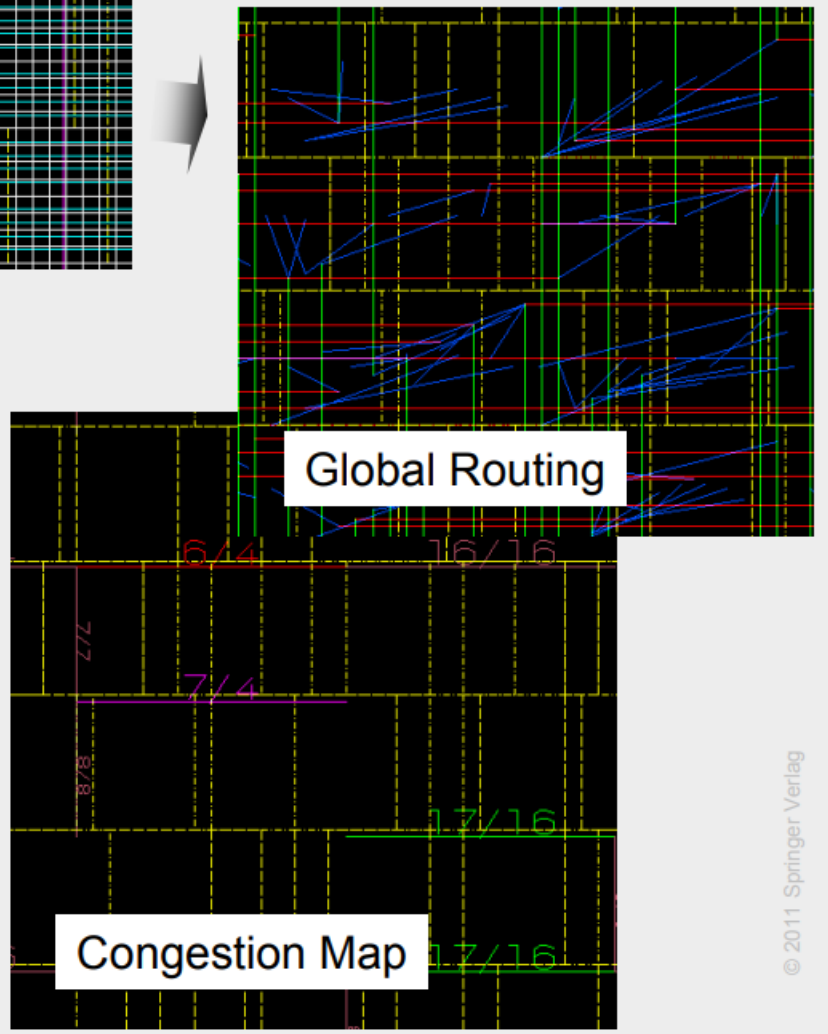
Detailed router tells us exact final path in this region

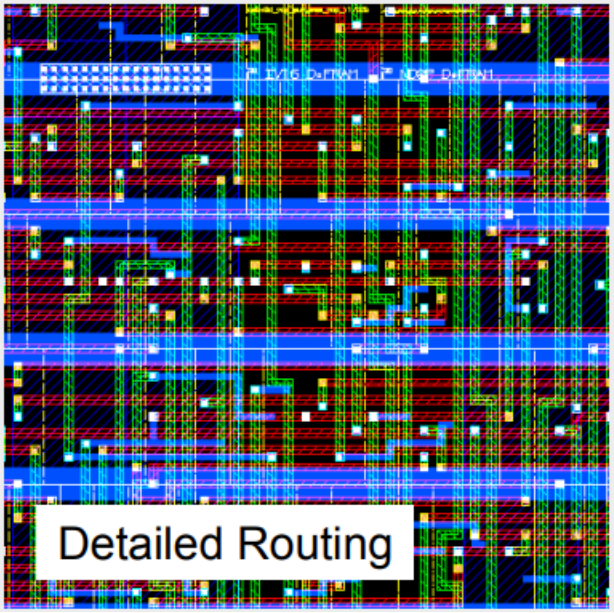


Too many overflows.
Not easy for DR



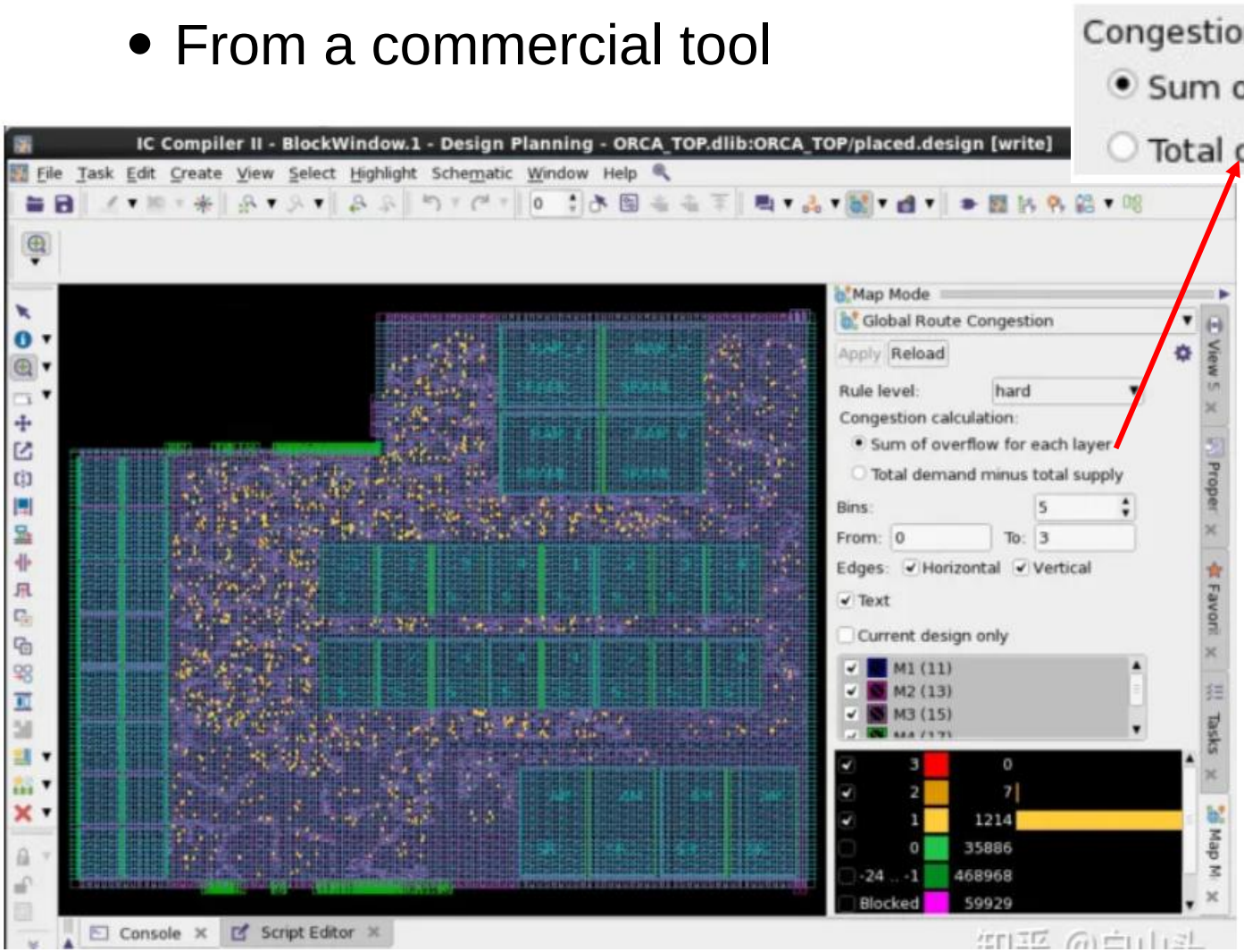


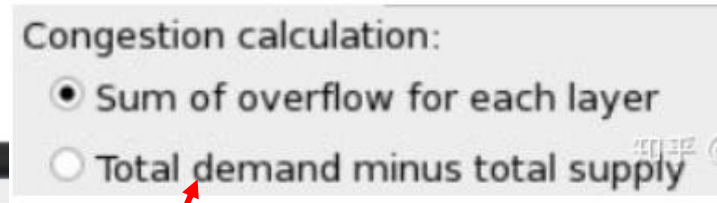




Congestion Map – An Important Indicator

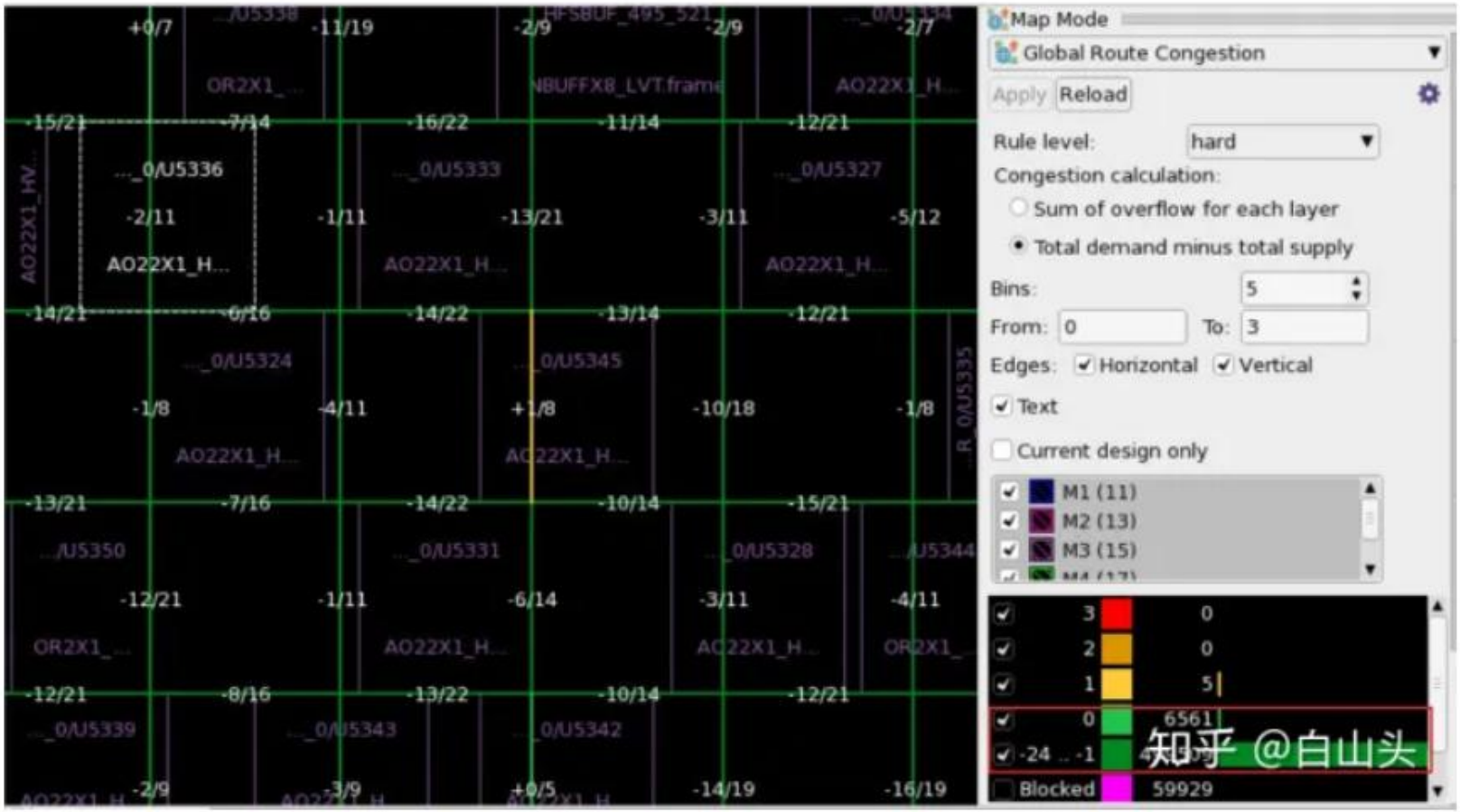
- From a commercial tool





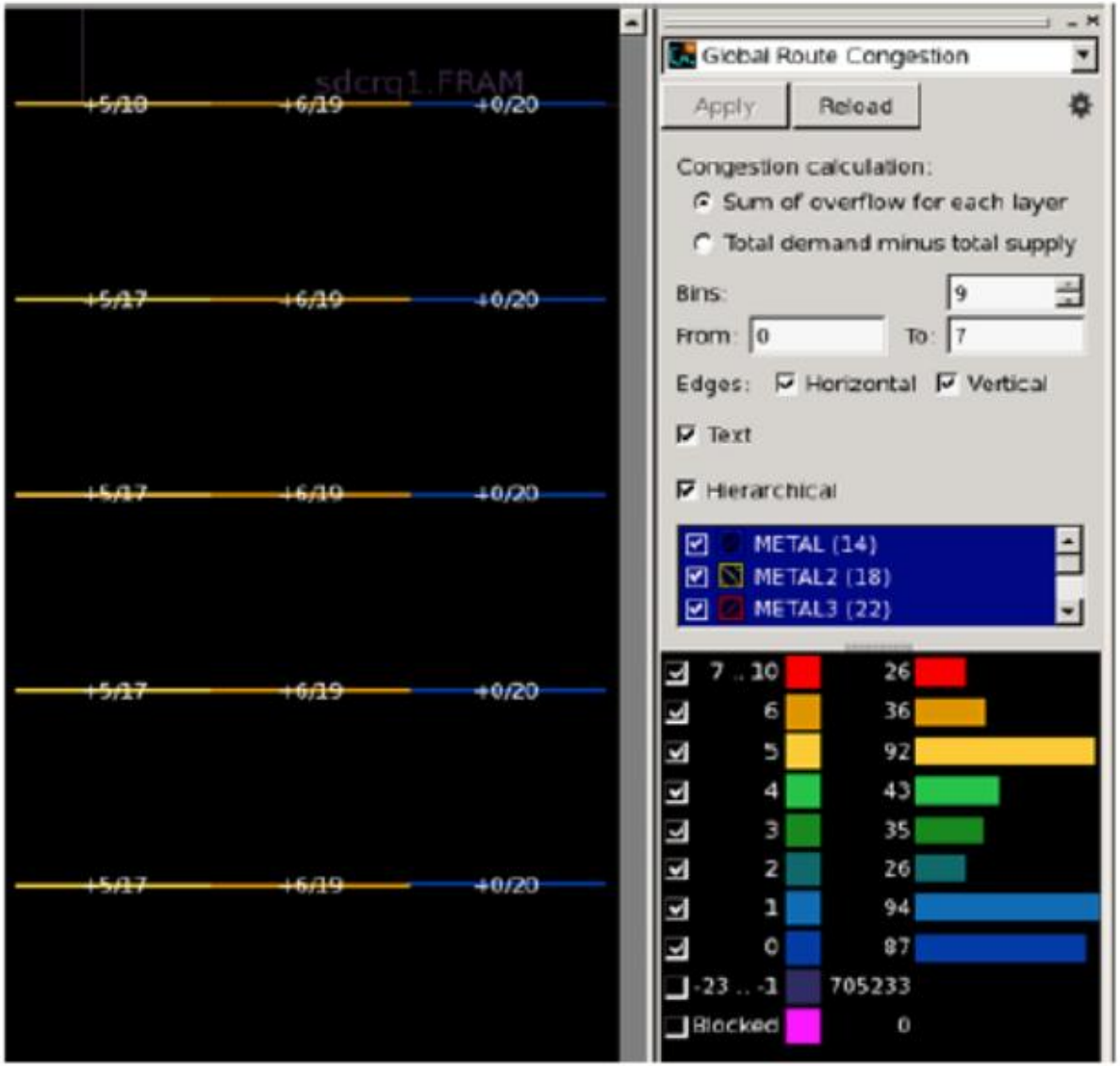
This shows this global router does layer assignment

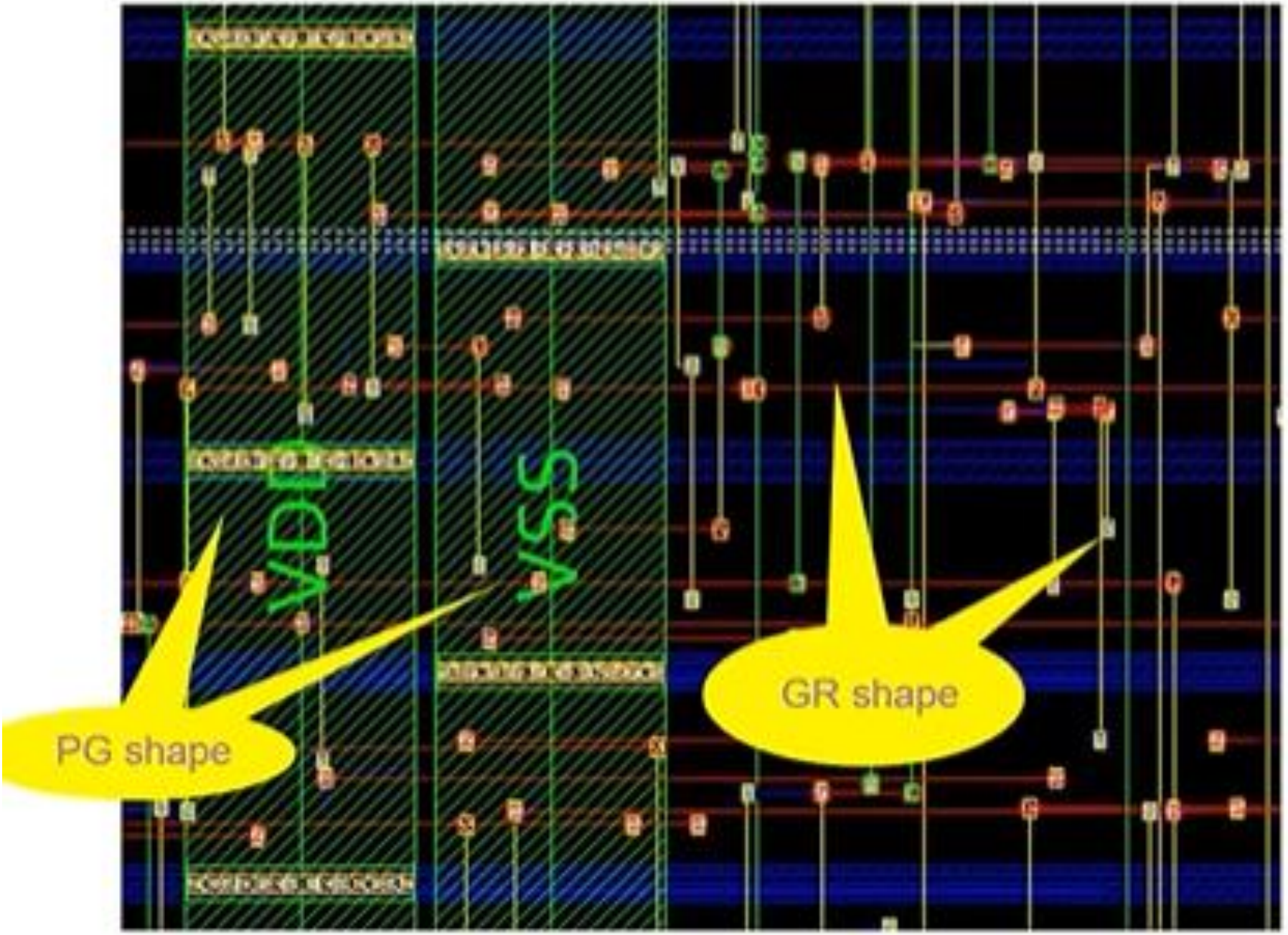
我们可以把所有的边都显示出来。



这样我们就能清楚的看到GRC的所有的边了。

Congestion Map



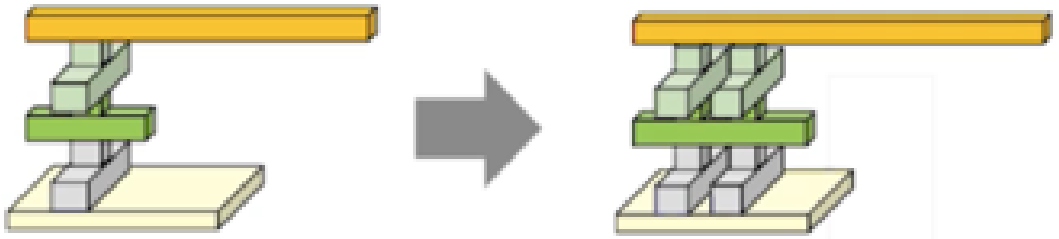


(For Tsing Hua University)

Via modeling and NDR will impact global routing

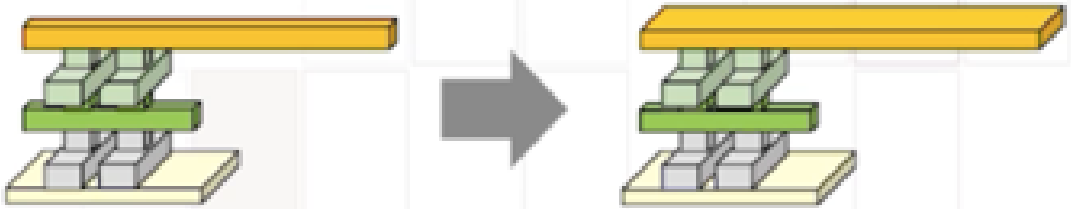
Via Pillar Insertion

Add parallel VIAs on source side of timing critical nets for R reduction



AutoNDR Routing

Enlarge metal width and spacing of timing critical nets (non-default routing, NDR) for RC reduction



Chapter 5 – Global Routing (taught at UCSD)

VLSI Physical Design: From Graph Partitioning to Timing Closure

Original Authors:

Andrew B. Kahng, Jens Lienig, Igor L. Markov, Jin Hu

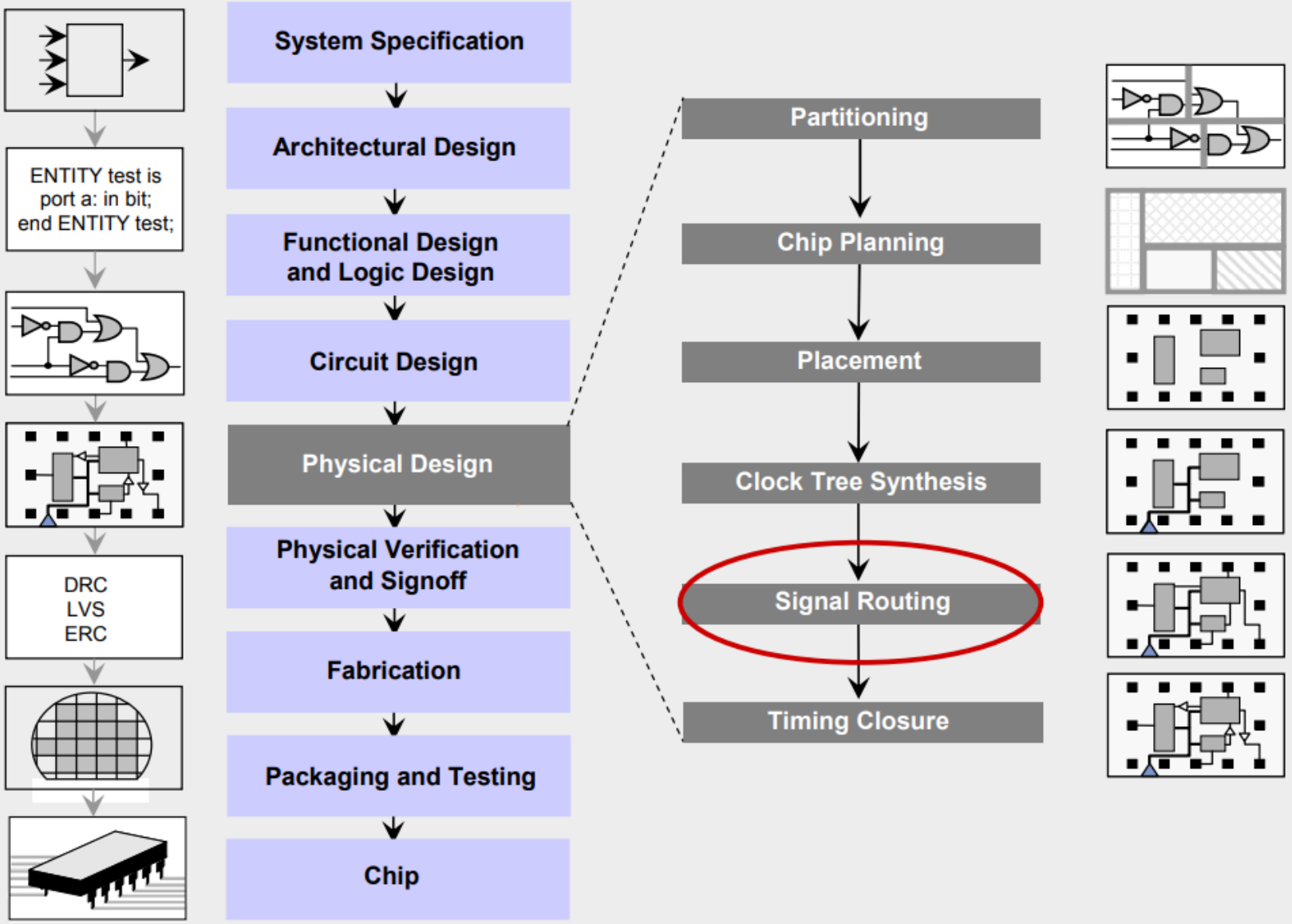
https://www.ifte.de/books/eda/chap5.pdf (2011)

Chapter 5 – Global Routing

5.3 Optimization Goals

- Global routing seeks to
  - determine whether a given placement is routable, and
  - determine a coarse routing for all nets within available routing regions
- Considers goals such as
  - minimizing total wirelength, and
  - reducing signal delays on critical nets

5.1 Introduction



5.1 Introduction

Given a placement, a netlist and technology information,

- determine the necessary wiring, e.g., net topologies and specific routing segments, to connect these cells
- while respecting constraints, e.g., design rules and routing resource capacities, and
- optimizing routing objectives, e.g., minimizing total wirelength and maximizing timing slack.

5.1 Introduction

Terminology:

- Net: Set of two or more pins that have the same electric potential
- Netlist: Set of all nets.
- Congestion: Where the shortest routes of several nets are incompatible because they traverse the same tracks.
- Fixed-die routing: Chip outline and routing resources are fixed.
- Variable-die routing: New routing tracks can be added as needed.

5.1 Introduction: General Routing Problem

Netlist:

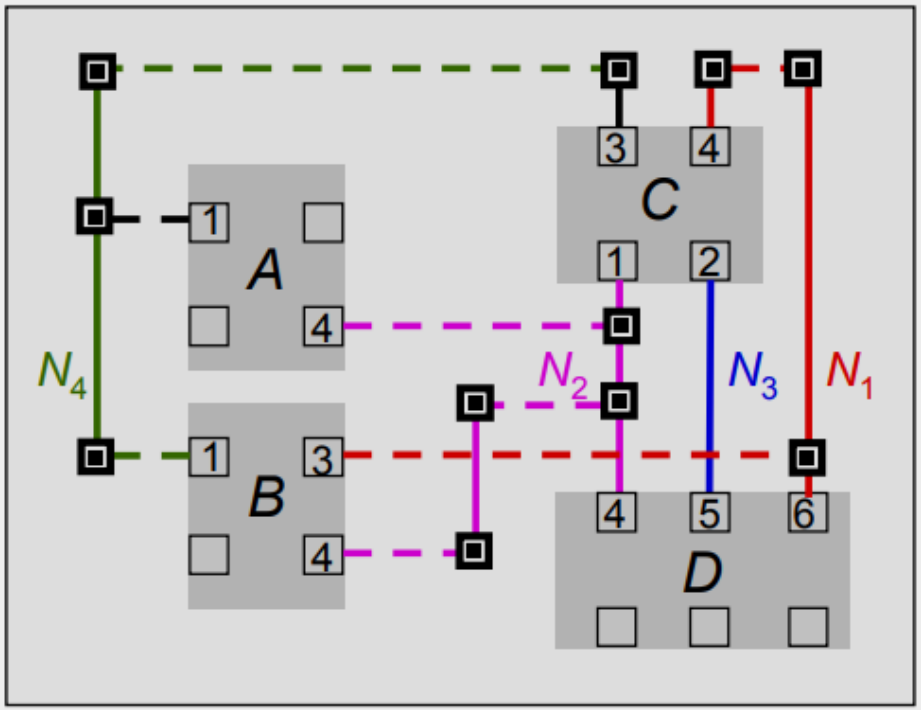
$N_1 = \{C_4, D_6, B_3\}$

$N_2 = \{D_4, B_4, C_1, A_4\}$

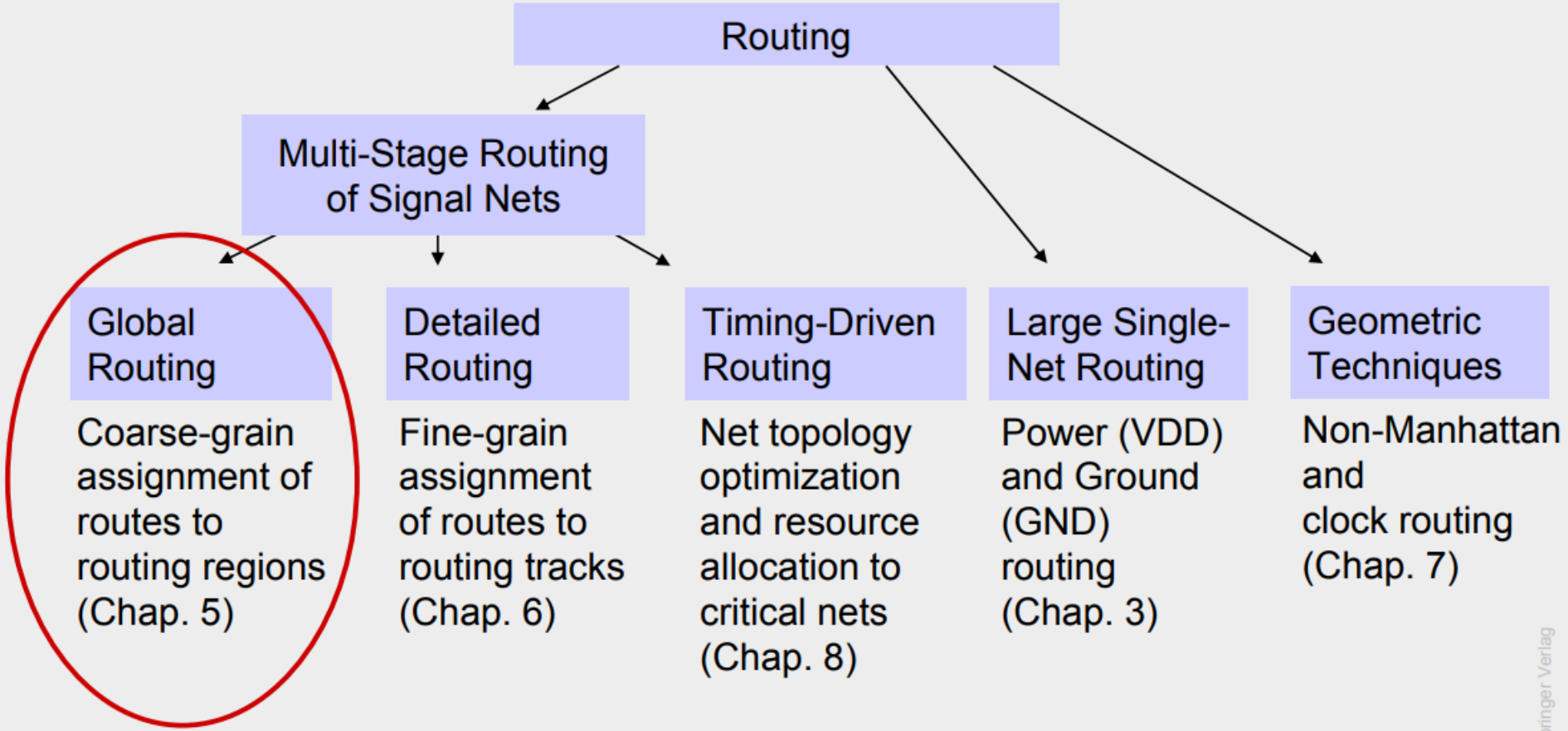
$N_3 = \{C_2, D_5\}$

$N_4 = \{B_1, A_1, C_3\}$

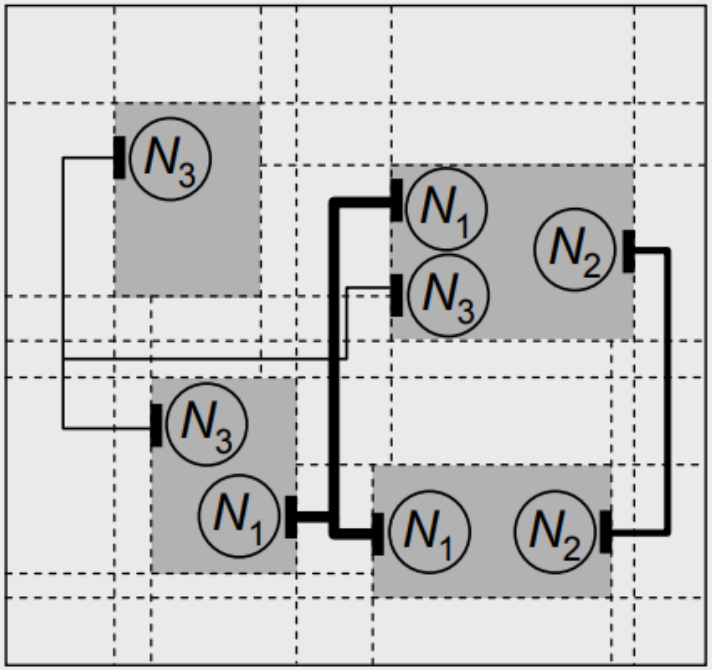
Technology Information
(Design Rules)



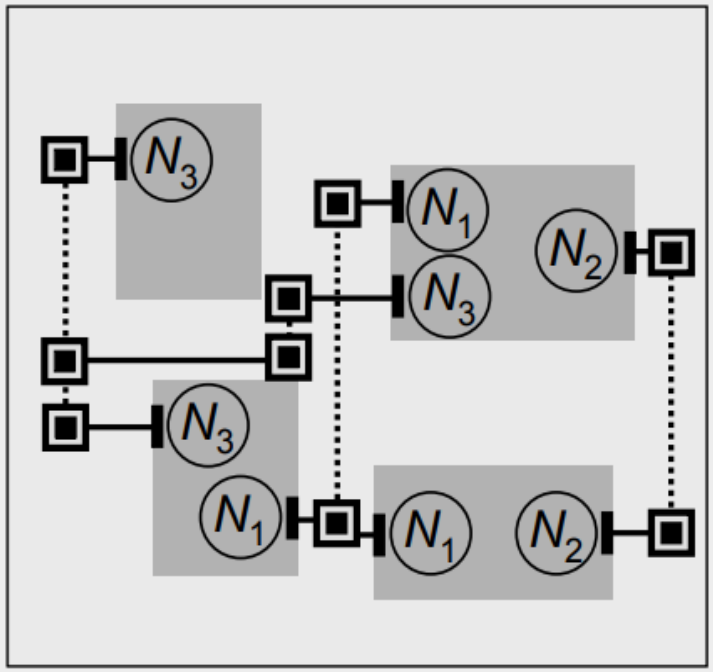
5.1 Introduction



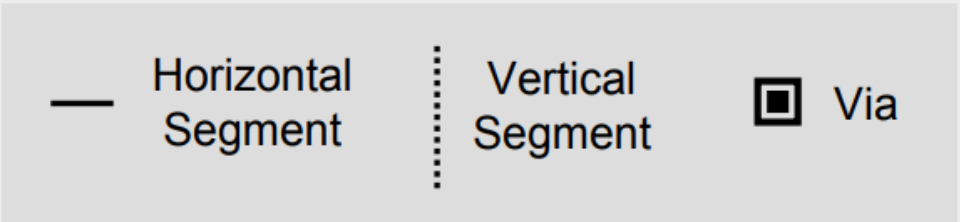
Global Routing



Detailed Routing

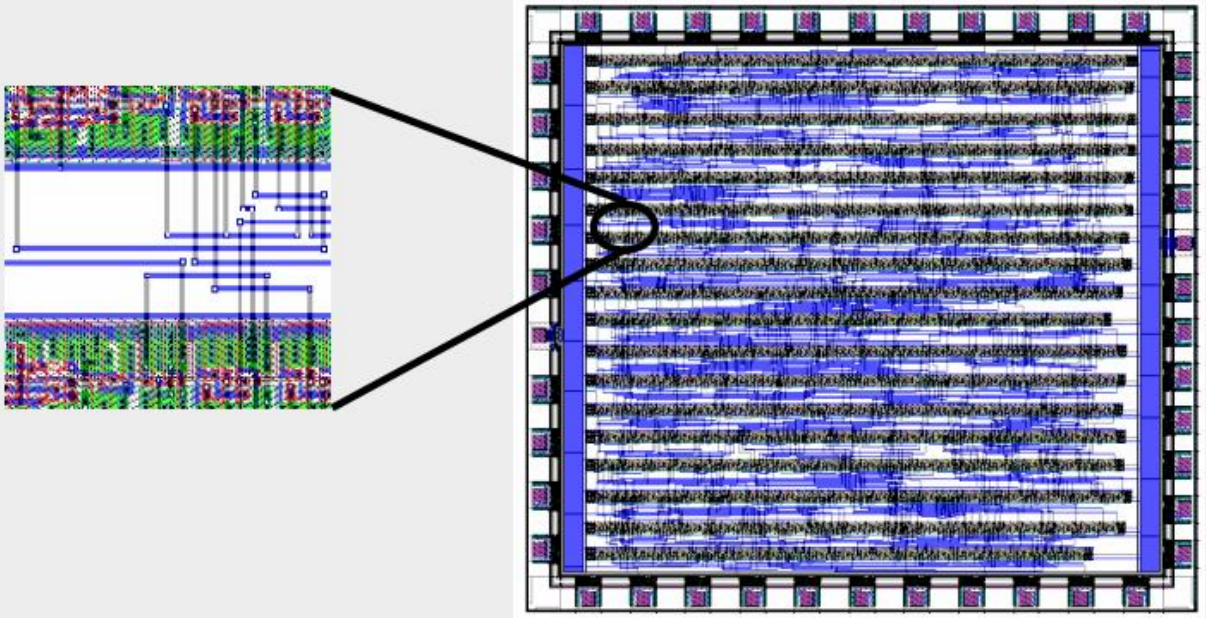


Channel connectivity graph. But if we can route over the macros, …



Channel

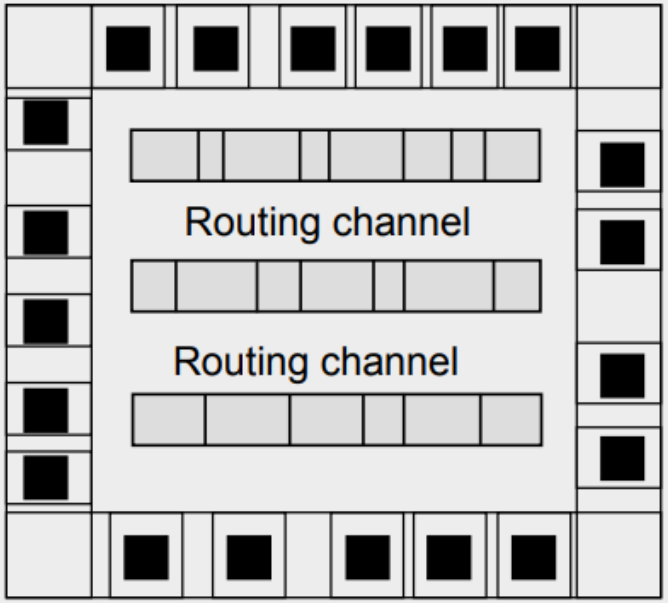
Rectangular routing region with pins on two opposite sides

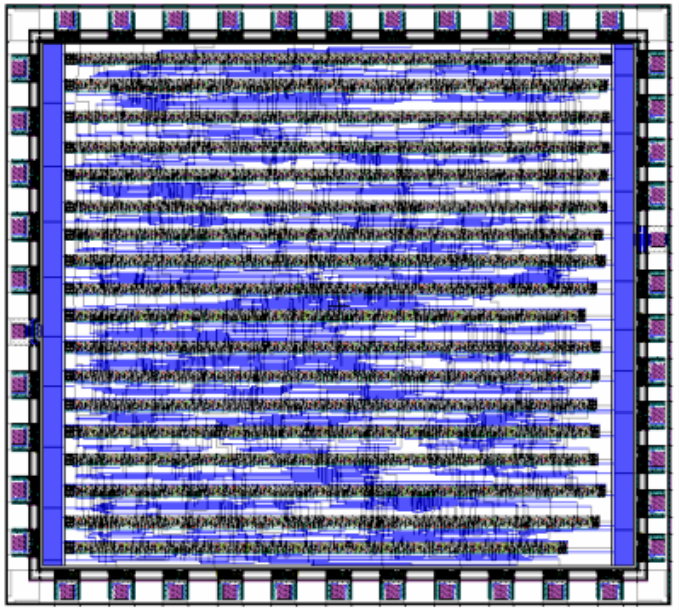


Standard cell layout (Two-layer routing)

Channel

Rectangular routing region with pins on two opposite sides





Standard cell layout (Two-layer routing)

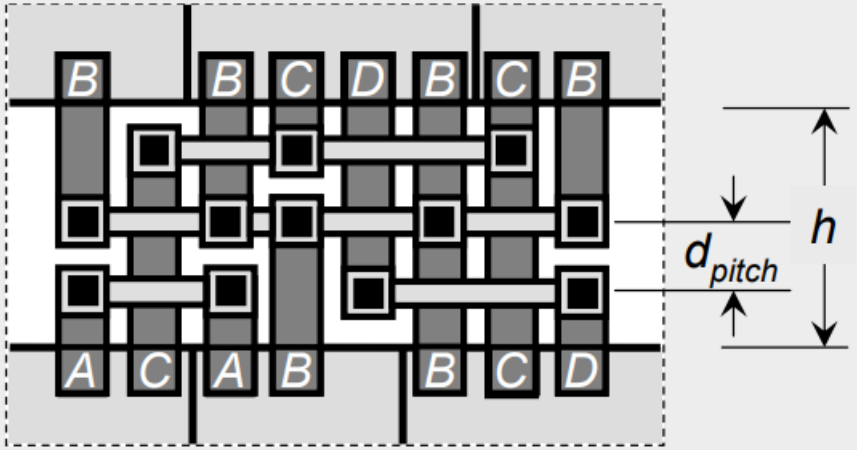
Capacity

Number of available routing tracks or columns

- For single-layer routing, the capacity is the height *h* of the channel divided by the pitch d_{pitch}
- For multilayer routing, the capacity σ is the sum of the capacities of all layers.

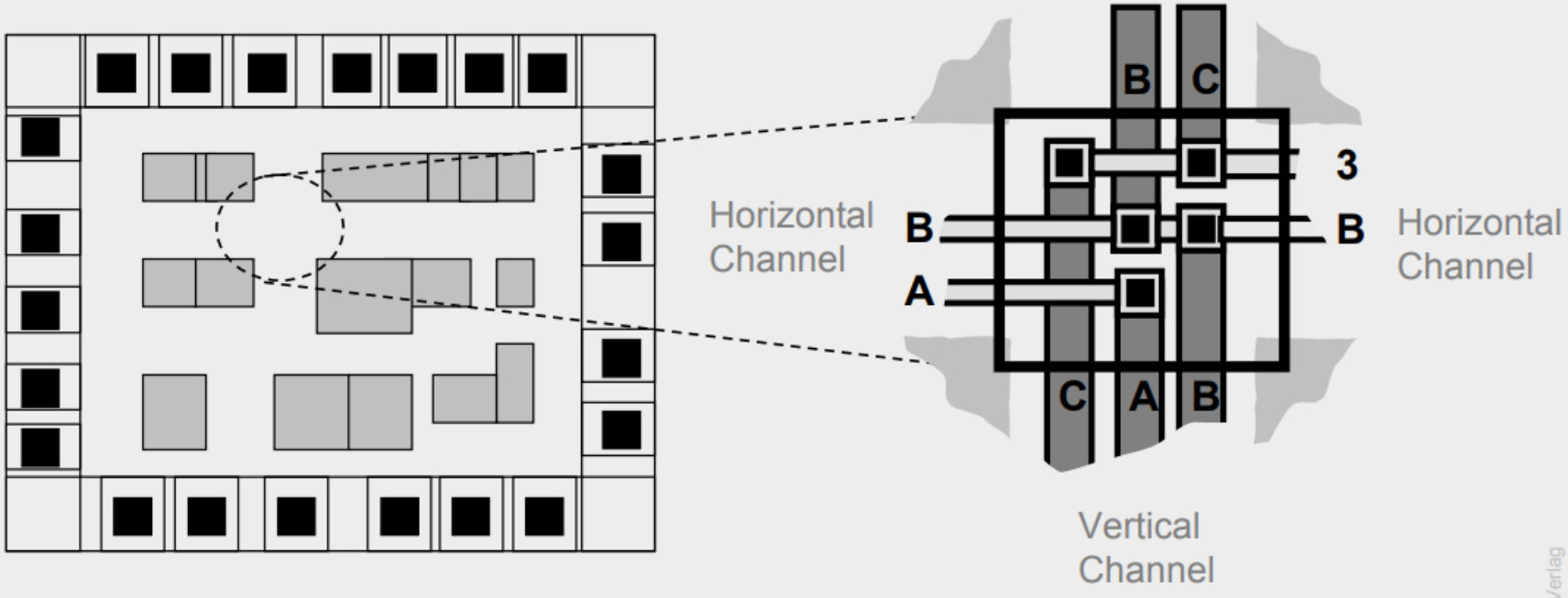
$$\sigma(Layers) = \sum_{layer \in Layers} \left\lfloor \frac{h}{d_{pitch}(layer)} \right\rfloor$$

Horizontal Routing Channel



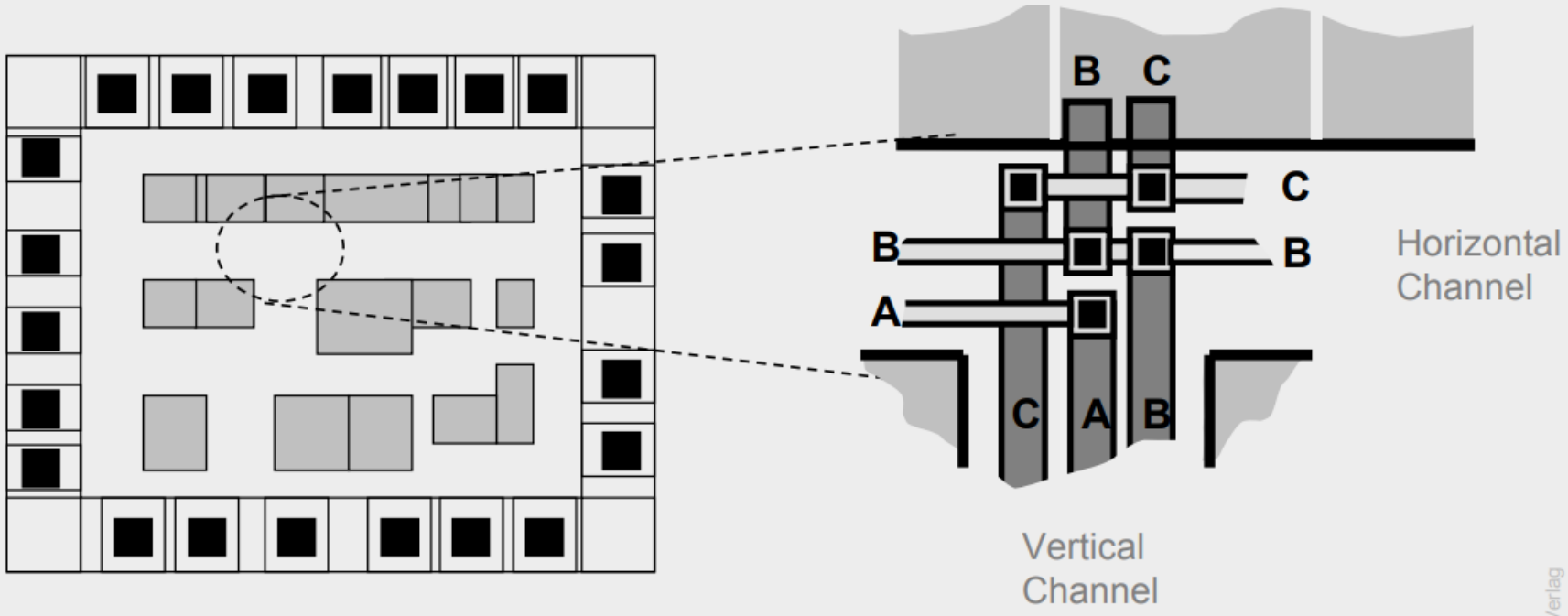
Switchbox (Two-layer macro cell layout)

Intersection of horizontal and vertical channels

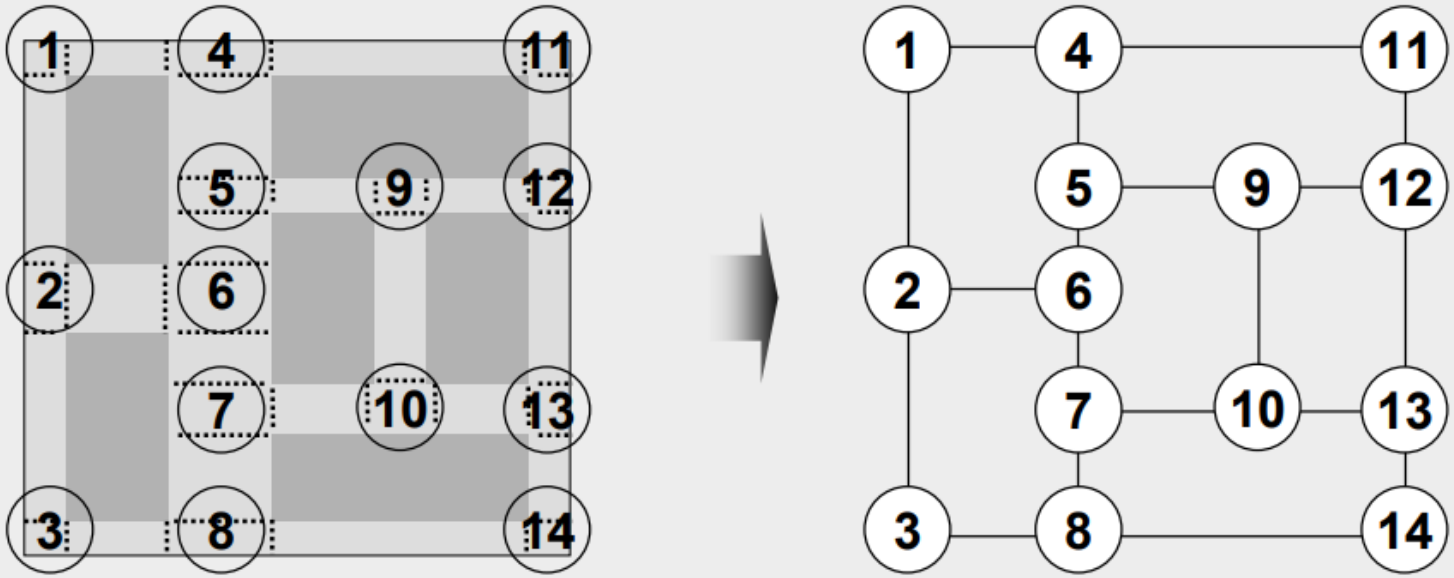


Horizontal channel is routed after vertical channel is routed

T-junction (Two-layer macro cell layout)



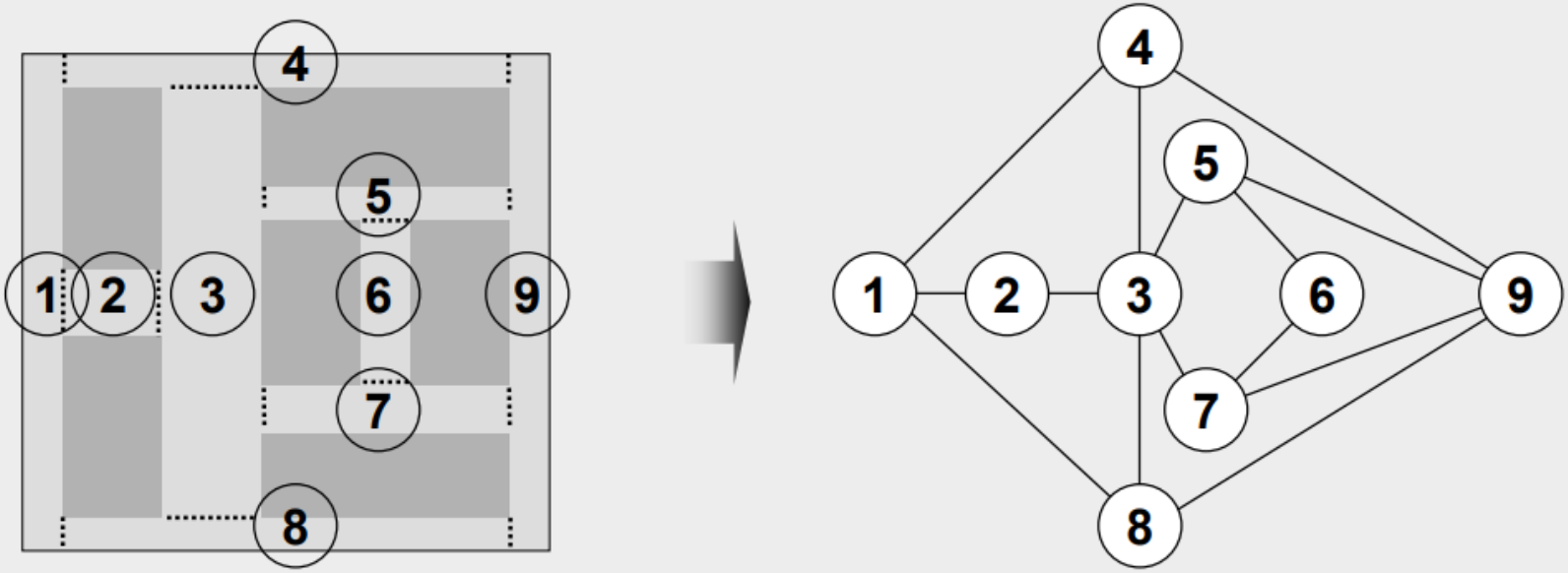
Switchbox connectivity graph



$G = (V, E)$, where the nodes $v \in V$ represent switchboxes
and an edge exists between two nodes if the corresponding switchboxes are on opposite sides of the same channel

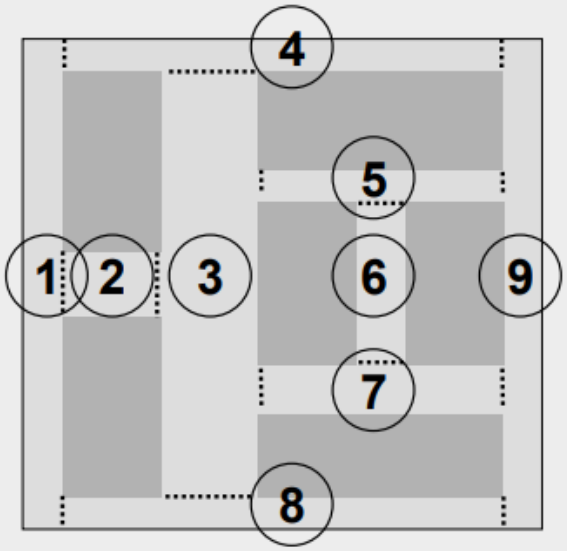
Channel connectivity graph

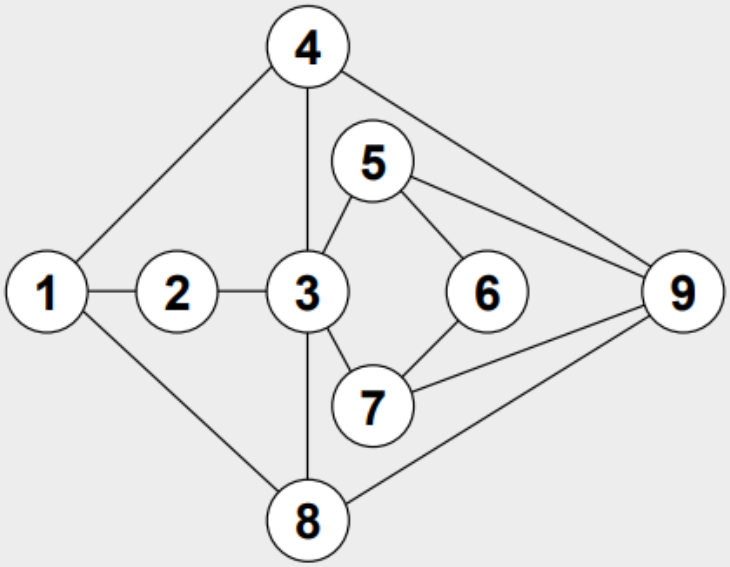
Over-the-macro routing is not allowed



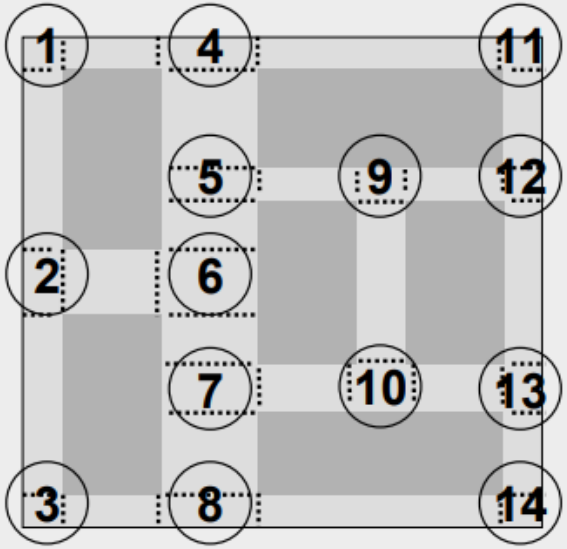
$G = (V,E)$, where the nodes $v \in V$ represent channels, and the edges E represent adjacencies of the channels

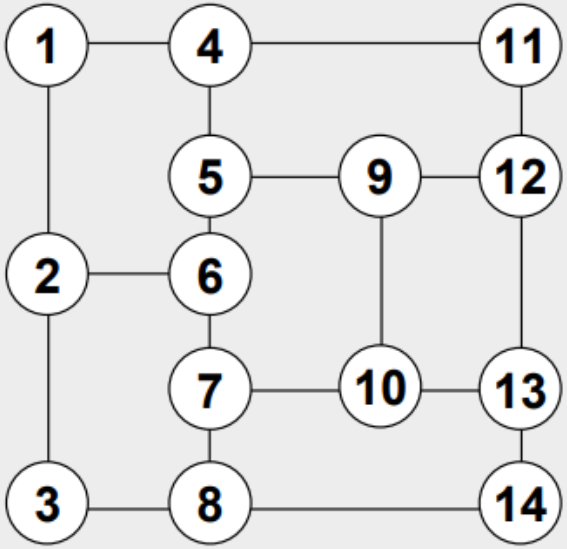
Channel connectivity graph

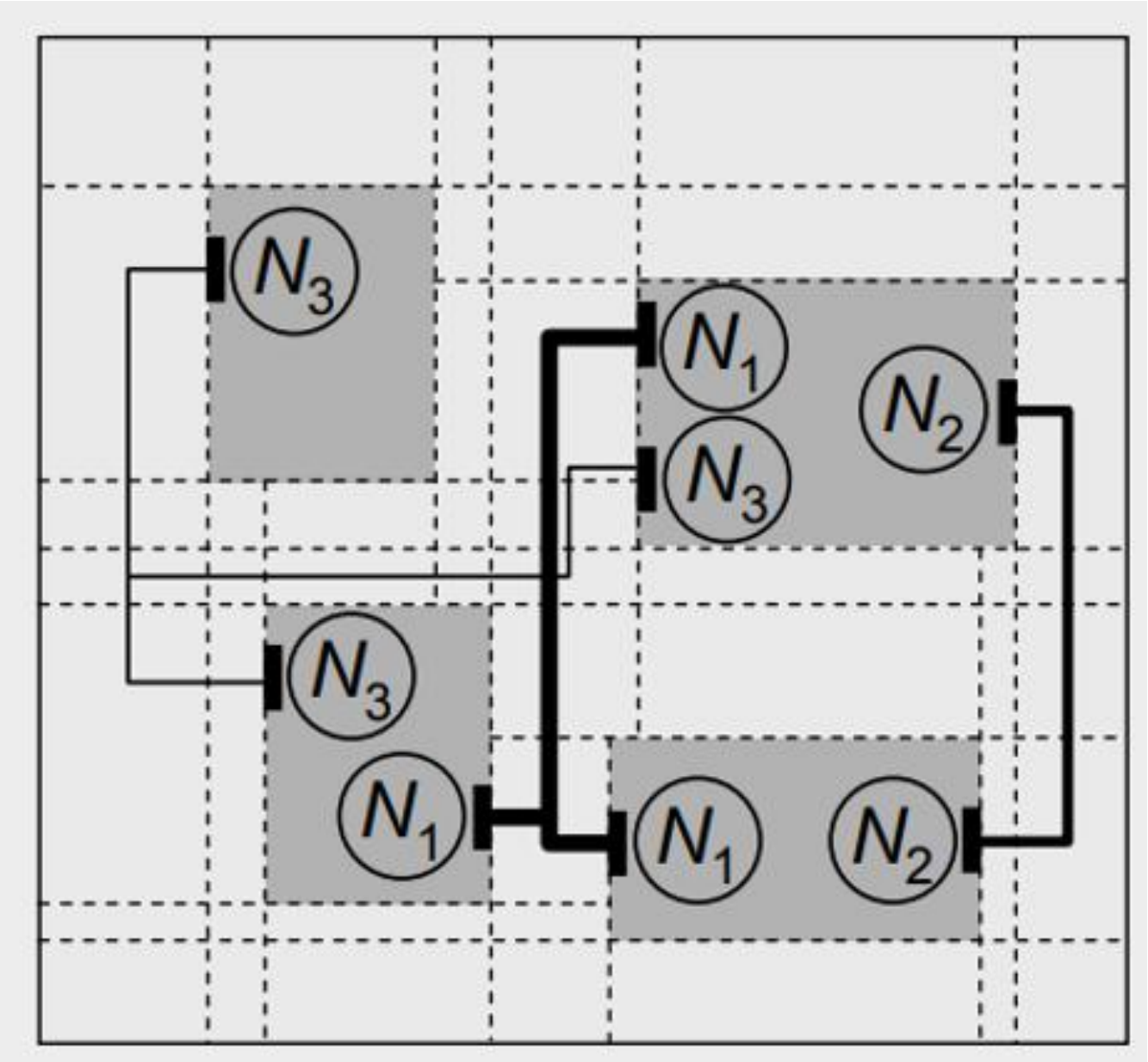




Switchbox connectivity graph






N_3
N_1
N_2
N_3
N_3
N_1
N_1
N_2

Global Routing Flow

- Creating Gcells (by superimposing the grids)
- Getting the routing tracks, blockages, pins within Gcells
  - Setting up capacities at each Gcell boundary
- Let each net get the shortest paths
  - Steiner tree, …
- There will be routing overflows at some Gcell borders
  - (i.e., demand is larger than capacity)
- Re-route within the GR graph to resolve overflows
  - Iterations; some slacks can be adjusted at Gcell borders
- Critical nets need to get shorter wirelengths
- Q1: if the global routing result does not have any overflow, will the detailed router complete the routing?
- Q2: if there are some overflows, can DR still finish the routing?

(For Tsing Hua University)

if the global routing result does not have any overflow, will the detailed router complete the routing?

(For Tsing Hua University)

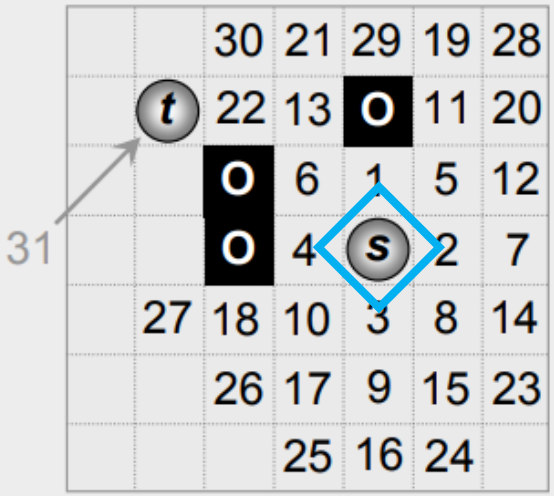
if there are some overflows, can DR still finish the routing?

https://drive.google.com/file/d/1hvzYdfBofFZrve1xyQz4giQOtt7Xr98B/view

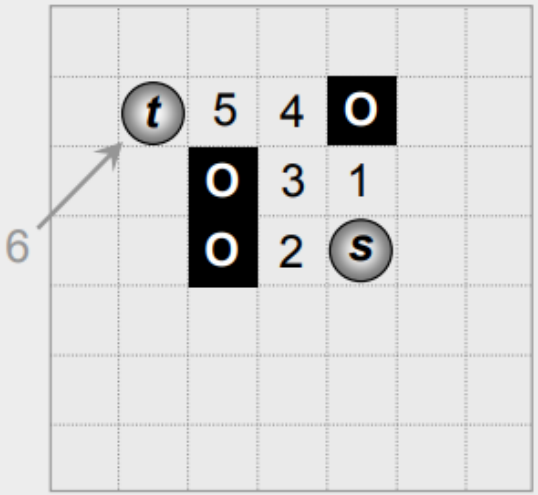
ICCAD 2024 Contest:
Problem D: Chip Level Global Router
聯發科技(Mediatek)

Next, we talk about how to route each nets? Or a group of nets?
Go back to slide-2, then, slide-11, 12

- A* search operates similarly to Dijkstra's algorithm, but extends the cost function to include an estimated distance from the current node to the target
- Expands only the most promising nodes; its best-first search strategy eliminates a large portion of the solution space



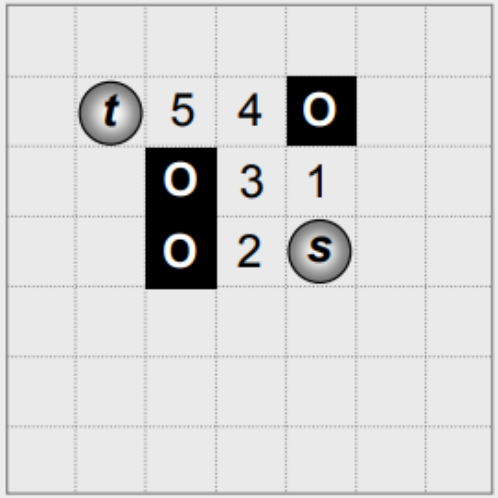
Dijkstra's algorithm
(exploring 31 nodes)



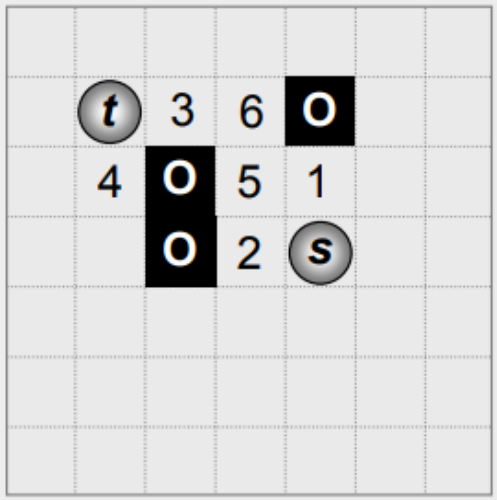
A* search
(exploring 6 nodes)

s Source
t Target
O Obstacle

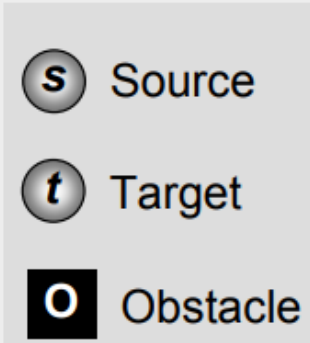
- Bidirectional A* search: nodes are expanded from both the source and target until the two expansion regions intersect
- Number of nodes considered can be reduced



Unidirectional A* search

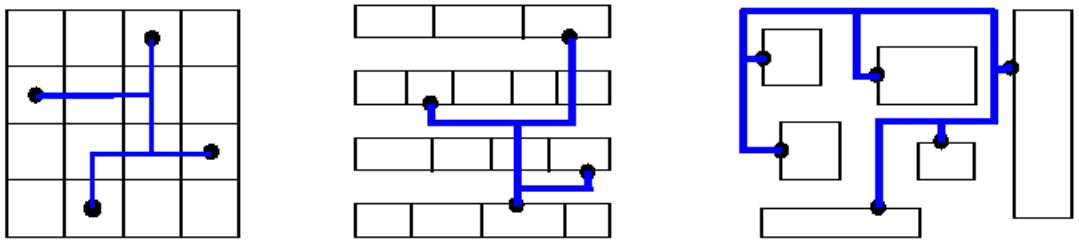


Bidirectional A* search

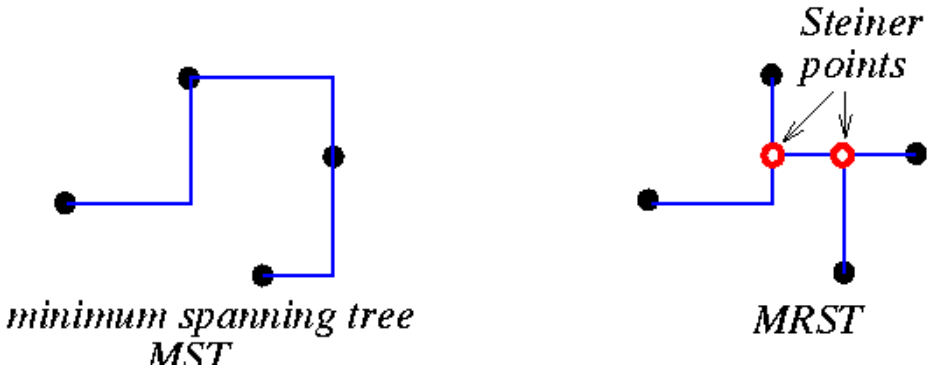


The Routing-Tree Problem

- **Problem:** Given a set of pins of a net, interconnect the pins by a "routing tree."

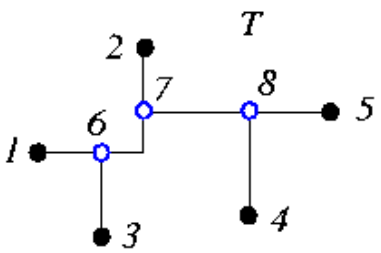


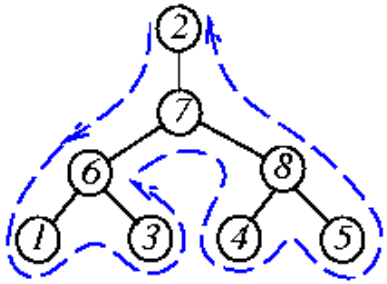
- **Minimum Rectilinear Steiner Tree (MRST) Problem:** Given n points in the plane, find a minimum-length tree of rectilinear edges which connects the points.
- $MRST(P) = MST(P \cup S)$, (MST= min-spanning-tree) where P and S are the sets of original points and Steiner points, respectively.



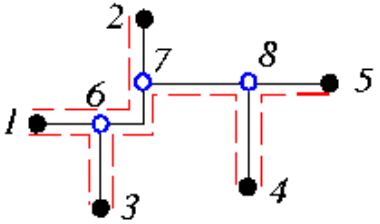
A Simple Performance Bound

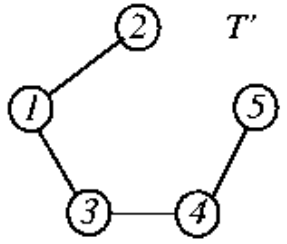
- Easy to show that $\frac{Cost(MST(P))}{Cost(MRST(P))} \leq 2$
- Given any MRST T on point set P with Steiner point set S, construct a spanning tree T' on P as follows:
  1. Select any point in T as a root.
  2. Perform a depth-first traversal on the rooted tree T.
  3. Construct T' based on the traversal.





- *depth–first traversal*
- *every edge is visited twice*



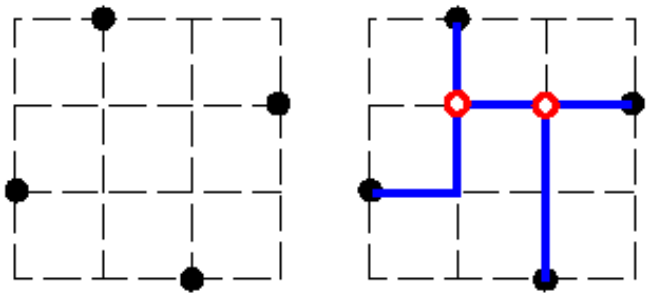


Cost(T') <= 2 Cost(T)

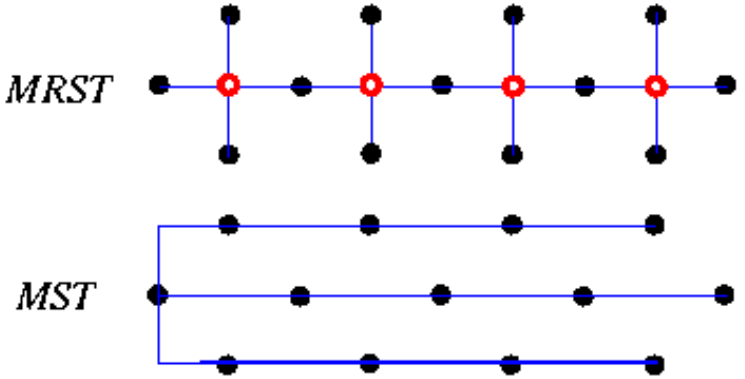
Every edge traversed twice

Theoretic Results for the MRST Problem

- **Hanan's Thm:** There exists an MRST with all Steiner points (set *S*) chosen from the **intersection** points of horizontal and vertical lines drawn points of *P*.
  - Hanan, "On Steiner's problem with rectilinear distance," *SIAM J. Applied Math.*, 1966.
- **Hwang's Theorem:** For any point set *P*, $\frac{Cost(MST(P))}{Cost(MRST(P))} \leq \frac{3}{2}$.
  - Hwang, "On Steiner minimal tree with rectilinear distance," *SIAM J. Applied Math.*, 1976.
- Best existing approximation algorithm: Performance bound 61/48 (=1.27) by Foessmeier *et al*.



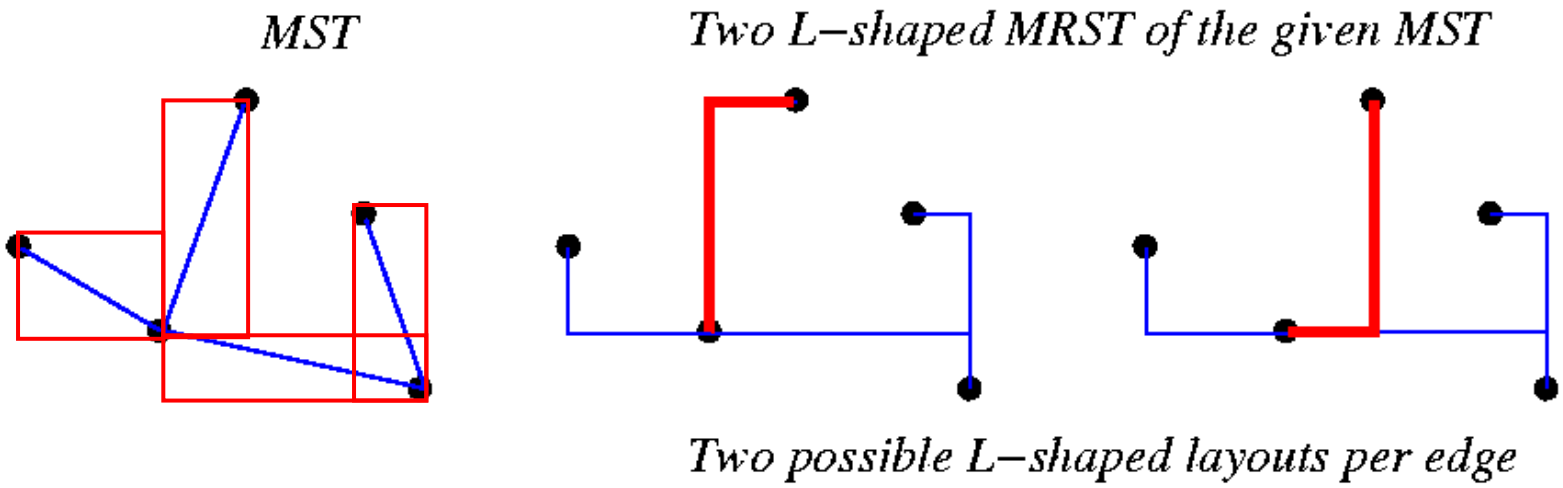
Hanan grid



Cost(MST)/Cost(MRST) –> 3/2

Coping with the MRST Problem

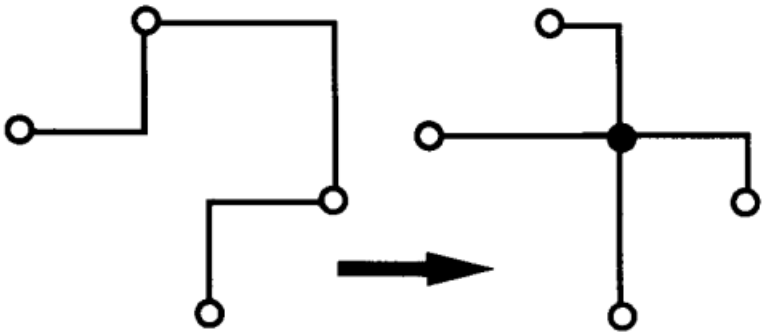
- Ho, Vijayan, Wong, “New algorithms for the rectilinear Steiner problem,”
  1. Construct an MRST from an MST.
  2. Each edge is straight or L-shaped.
  3. Maximize overlaps by dynamic programming.
- About 8% smaller than *Cost*(*MST*).



Iterated 1-Steiner Heuristic for MRST

- Kahng & Robins, “A new class of Steiner tree heuristics with good performance: the iterated 1-Steiner approach,” *ICCAD-90.*.

Intuitively we ask: if we are allowed to use **one** Steiner point, where will it be?



A 1-Steiner tree is the minimum spanning tree over (P U {x=Steiner pt.})

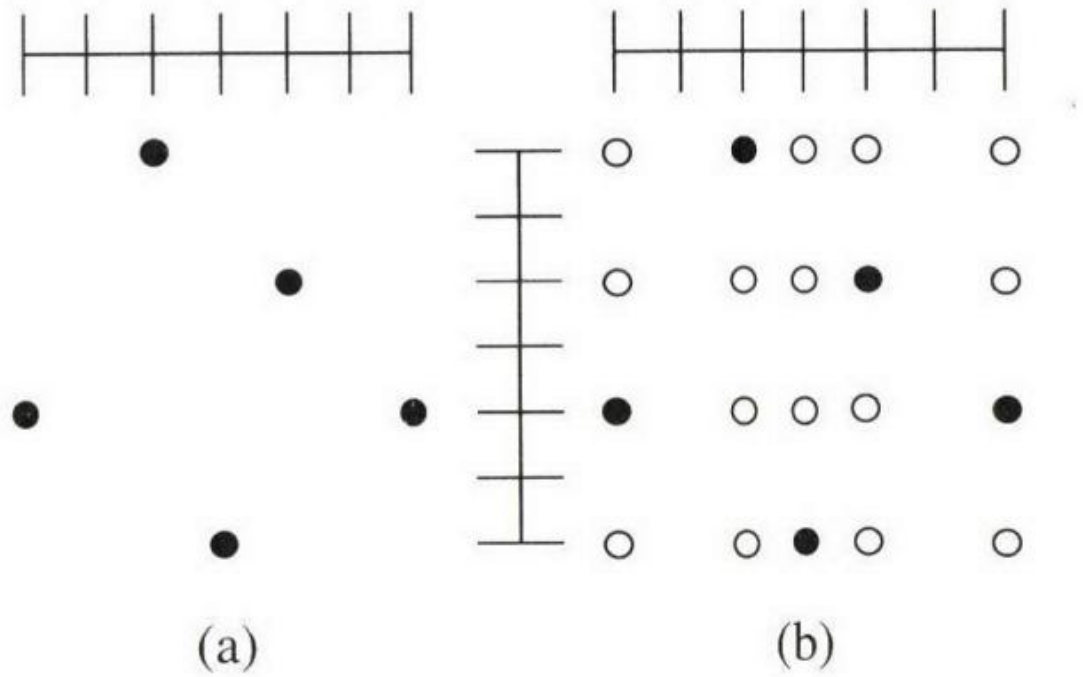
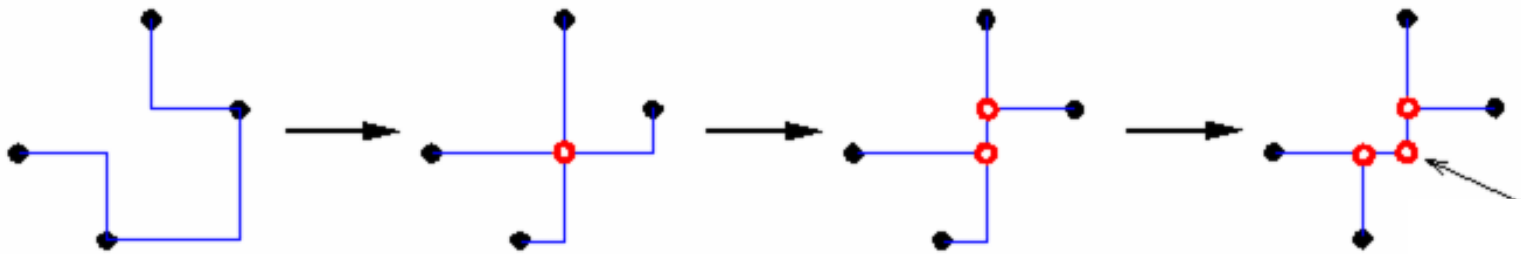


Figure 9.20 A set of points P for which the Steiner-tree should be constructed (a) and its Hanan points (b).

Algorithm: Iterated_1-Steiner(*P*)
***P*: set *P* of *n* points. (Note: result can have more Steiner points)**
1 begin
2 $S \leftarrow \emptyset$;
/* $H(P \cup S)$: set of Hanan points */
/* $\Delta MST(A, B) = Cost(MST(A)) - Cost(MST(A \cup B))$ */

3 while ($Cand \leftarrow \{x \in H(P \cup S) | \Delta MST(P \cup S, \{x\}) > 0\} \neq \emptyset$) do
4 Find $x \in C$ and which maximizes $\Delta MST(P \cup S), \{x\})$;
5 $S \leftarrow S \cup \{x\}$;
6 Remove degenerate points in *S* which have degree ≤ 2 in $MST(P \cup S)$;
7 Output $MST(P \cup S)$;
8 end



https://vlsicad.ucsd.edu/Publications/Journals/j4.pdf

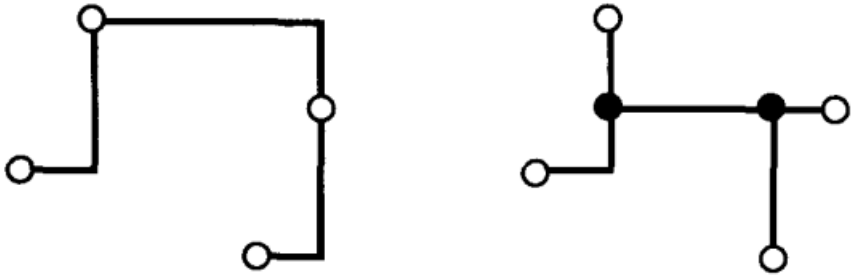


Fig. 1. MST (*left*) and MRST (*right*) for the same four-point set.

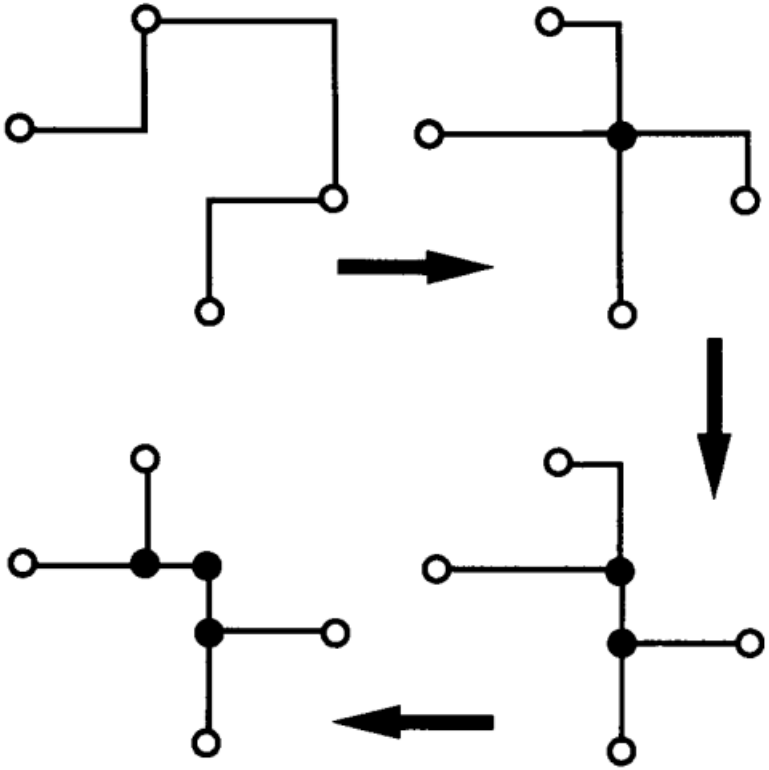


Fig. 3. Execution of iterated 1-Steiner on a four-point example.

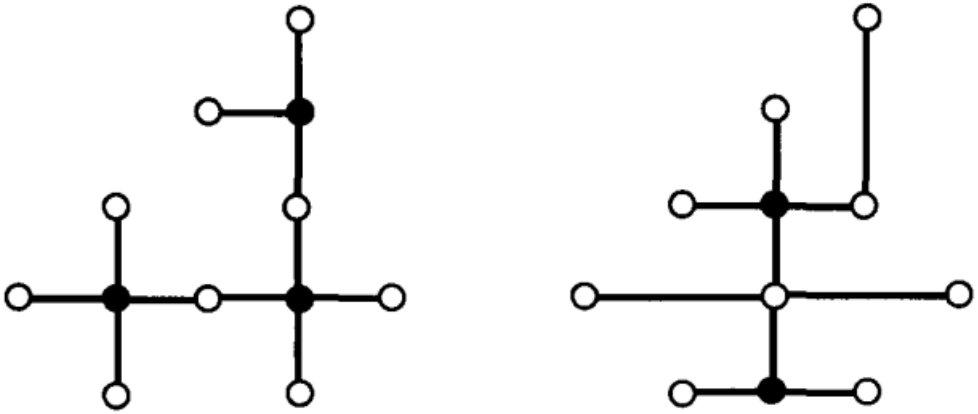


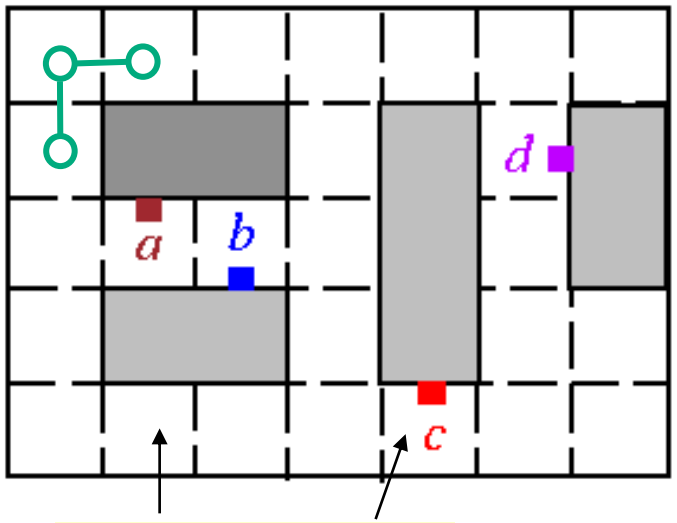
Fig. 8. A nine-point example where the iterated 1-Steiner performance ratio is 13/11; the optimal MRST (*left*) has cost 11, while the (possible) heuristic output (*right*) has cost 13.

Global Routing

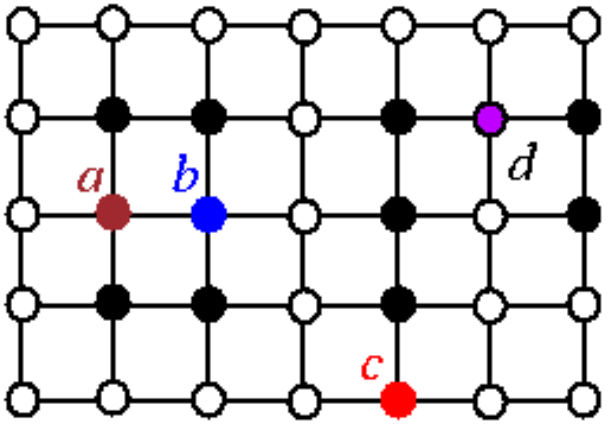
- Global Routing Problem Formulation
- Single Net Routing
  - Spanning Tree
  - Steiner Tree
  - Rectilinear Steiner Tree
- Routing All Nets
  - Iterative Improvement
  - Negotiation Based Routing
  - Iterative Deletion
  - Multi-commodity Flow Based Routing

Graph Models for Global Routing: Grid Graph

- Each cell is represented by a vertex.
- Two vertices are joined by an edge if the corresponding cells are adjacent to each other.
- The occupied cells are represented as filled circles, whereas the others are as clear circles.



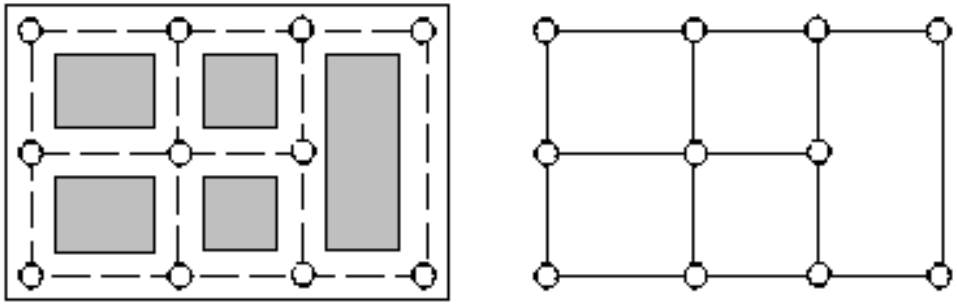
Global routing cells - Gcells



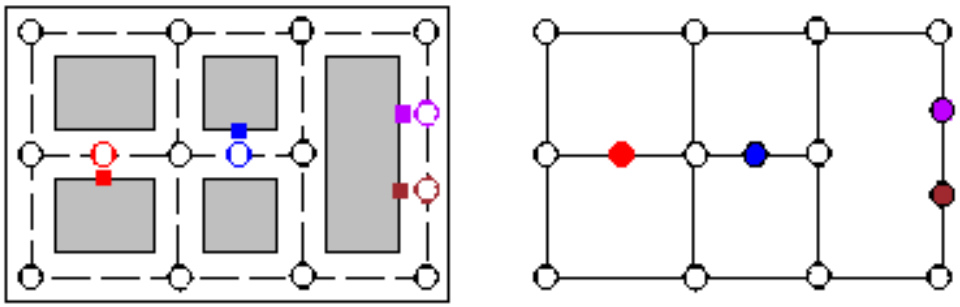
Graph Model: Channel Intersection Graph

- Channels are represented as edges.
- Channel intersections are represented as vertices.
- Edge weight represents channel capacity.
- Extended channel intersection graph: terminals are also represented as vertices.

channel intersection graph



extended channel intersection graph

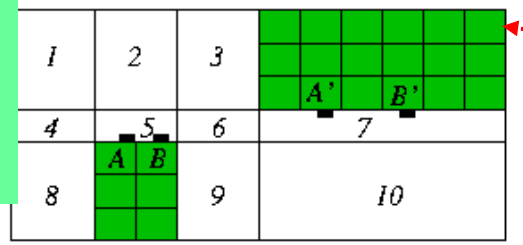


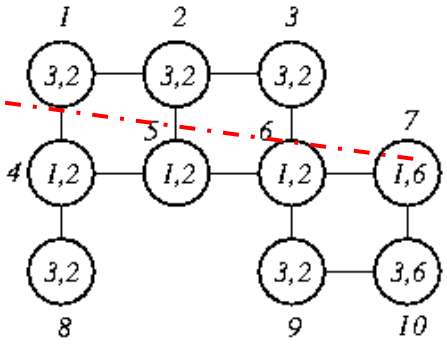
But if we can route over the blocks, how to model them?

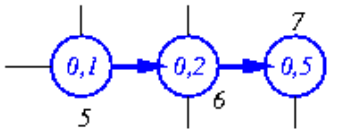
Global-Routing: Maze Routing

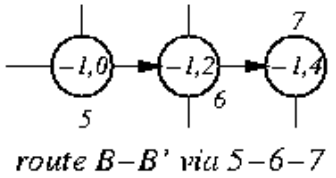
- Routing channels may be modelled by a weighted undirected graph called **channel connectivity graph**.
- Node ↔ channel; edge ↔ two adjacent channels; capacity: (*width*, *length*)

Each green is one length. Cell-1 has 3 horizontal tracks, 2 vertical tracks



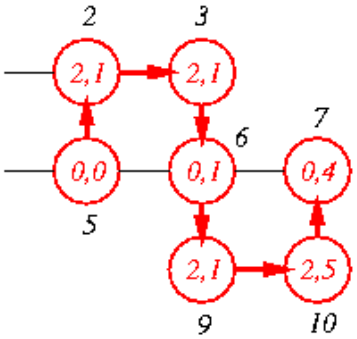




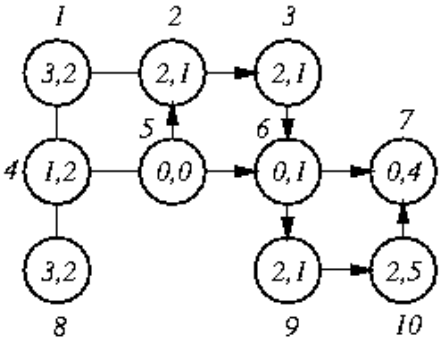


B can't go thru 5-6-7, since negative

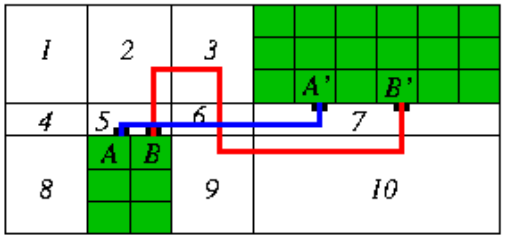
Gcell size is not uniform. It uses length to denote



route B–B' via 5–2–3–6–9–10–7



updated channel graph



maze routing for nets A and B

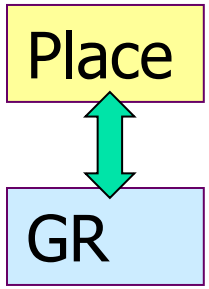
So Far We Covered

- Global routing graph modeling
- Global routing formulation
- High level algorithms
  - Each net is global routed by shortest paths
    - Using L-shapes improvement to approximate MRST
    - Every time adding one more Steiner point to approximate MRST
    - A* maze search
  - Net ordering?
  - Very likely some Gcell boundries have overflows
  - How about rip-up-and-reroute to smooth out congestions?

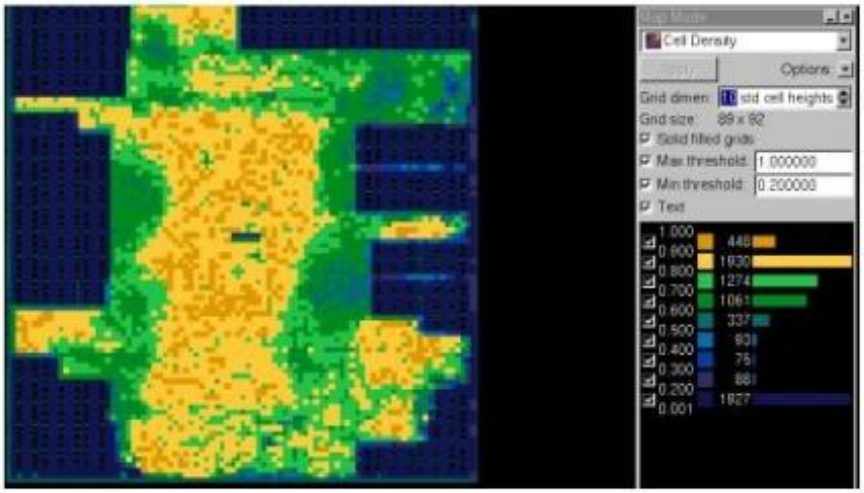
https://vlsi-backend-adventure.com/congestion_analysis.html

Congestion Fixes

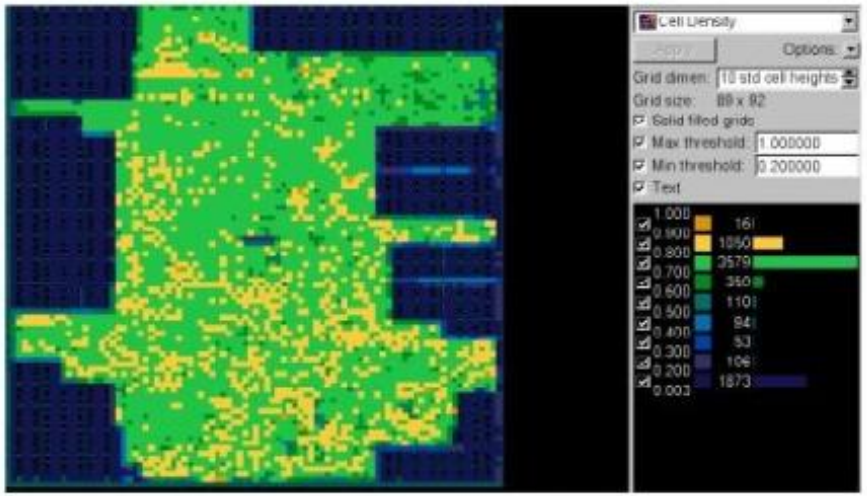
- Add placement blockages in channels and around macro corners
- Review the macro placement
- Reduce local cell density using density screens
- Reordering scan chain to reduce congestion
- Congestion driven placement with high effort
- Continue the iterations until good congestion results
- Density screen is applied to limit the density of standard cells in an area to reduce congestion due high pin density



Congestion Profiles

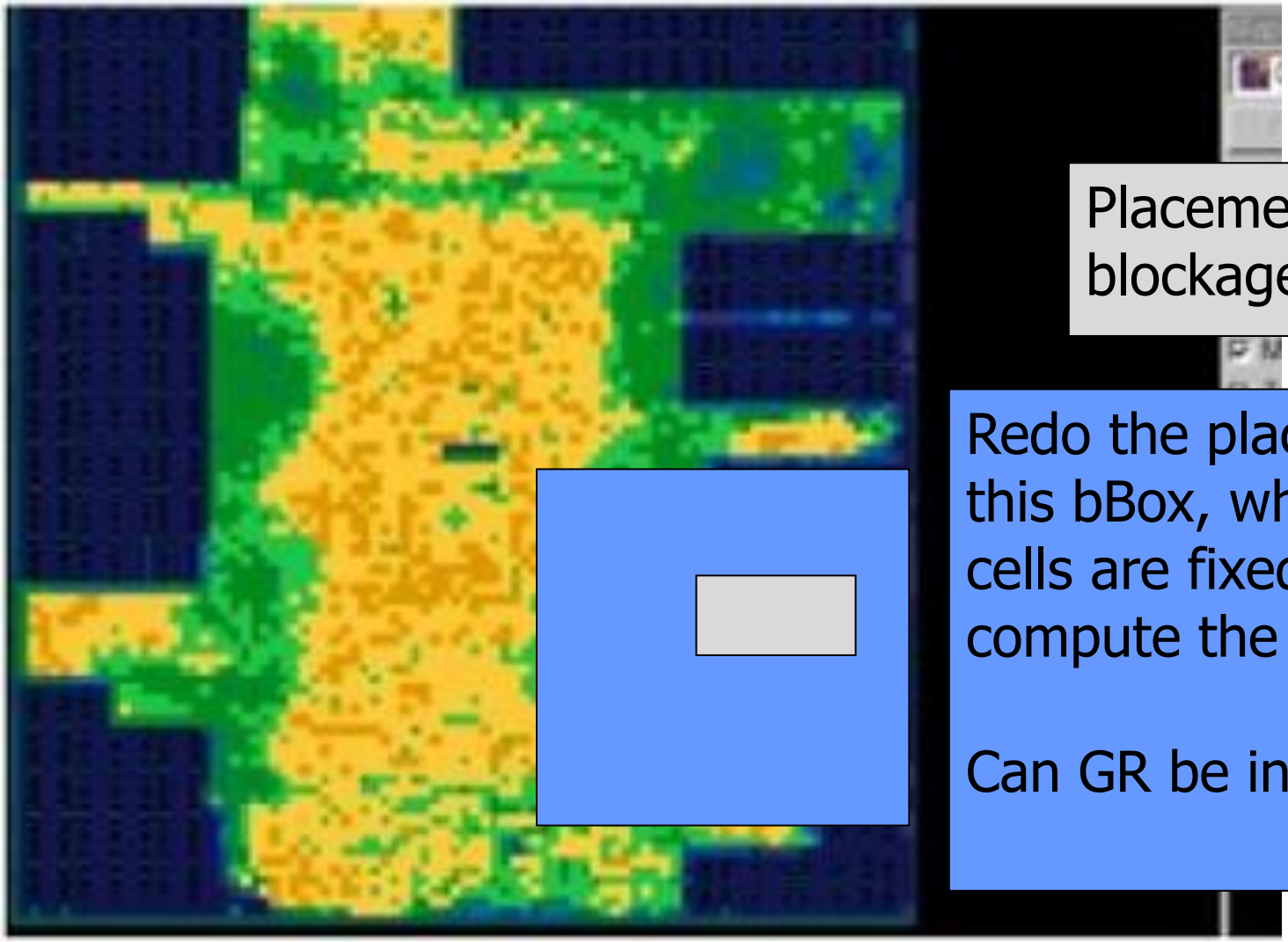


Before Fixing



After Fixing

Congestion Profiles



Redo the placement within this bBox, while all other cells are fixed. It first compute the utilization %

Can GR be incremental?

Before Fixing

Another Good Materials For Global Routing

https://www.ifte.de/books/eda/chap5.pdf (2011)

https://cadlab.cs.ucla.edu/~cong/slides/iccad-routing-tutorial_final_all.pdf

http://www.cecs.uci.edu/~papers/iccad08/PDFs/Papers/05A.1.pdf
NTHU-Route 2.0: A Fast and Stable Global Router

Abstract—We present in this paper a fast and stable global router called NTHU-Route 2.0 that improves the solution quality and runtime of a state-of-the-art router, NTHU-Route, by the following enhancements:
(1) a new history based cost function,
(2) new ordering methods for congested region identification and rip-up and reroute, and
(3) two implementation techniques.
It does 2D GR first, then, assign layers.

The experimental results show that NTHURouter 2.0 solves all ISPD98 benchmarks with very good quality. Moreover, it routes 7 of 8 ISPD07 benchmarks without any overflow. In particular, for one of the ISPD07 benchmarks which are thought to be difficult cases previously, NTHU-Route 2.0 can completely eliminate its total overflow. NTHU-Route 2.0 also successfully solves 12 of 16 ISPD08 benchmarks without causing any overflow.

Pictures From NTHU-Route 2.0

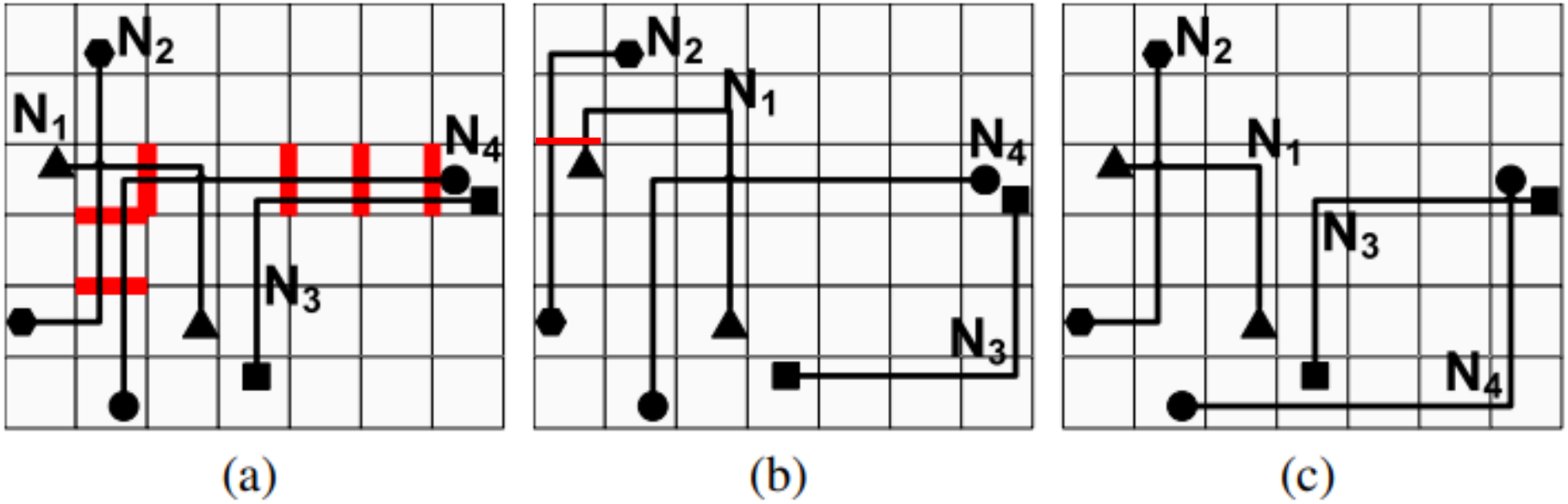


Fig. 5. (a) A routing solution obtained from a previous iteration. (in (a), N4 passes more congested edges.)
(b) A routing solution obtained by NTHU-Route.
(c) A routing solution obtained by NTHU-Route with the new ordering method for rip-up and reroute.

https://ieeexplore.ieee.org/document/4603083
Congestion, via minimization in layer assignment

GitHub

https://github.com/AUCOHL/OGRE

Global Router Built for ICCAD-Contest 2019 - Prerequisites

Make sure to first have G++ Compiler and Boost.

What things you need to do to get the Lef Def Parser working

(I saw Chris Chu's works were used.)

https://github.com/cuhk-eda/cu-gr (worthwhile)

CUGR is a detailed routability-driven global router and its solution quality is solely determined by the final detailed routing results. In particular, our global router adopts several efficient and effective methods to generate a set of connected rectangles to guide the detailed router:

- A sophisticated probability-based cost scheme
- An optimal 3D pattern routing technique that combines 2D pattern routing and layer assignment
- A multi-level maze routing utilizes two levels of routing
- A patching technique that adds useful route guides to further improve the detailed routability.

https://ieeexplore.ieee.org/document/9218646 (downloadable in school. Can be a choice for proj-4)

CUGR

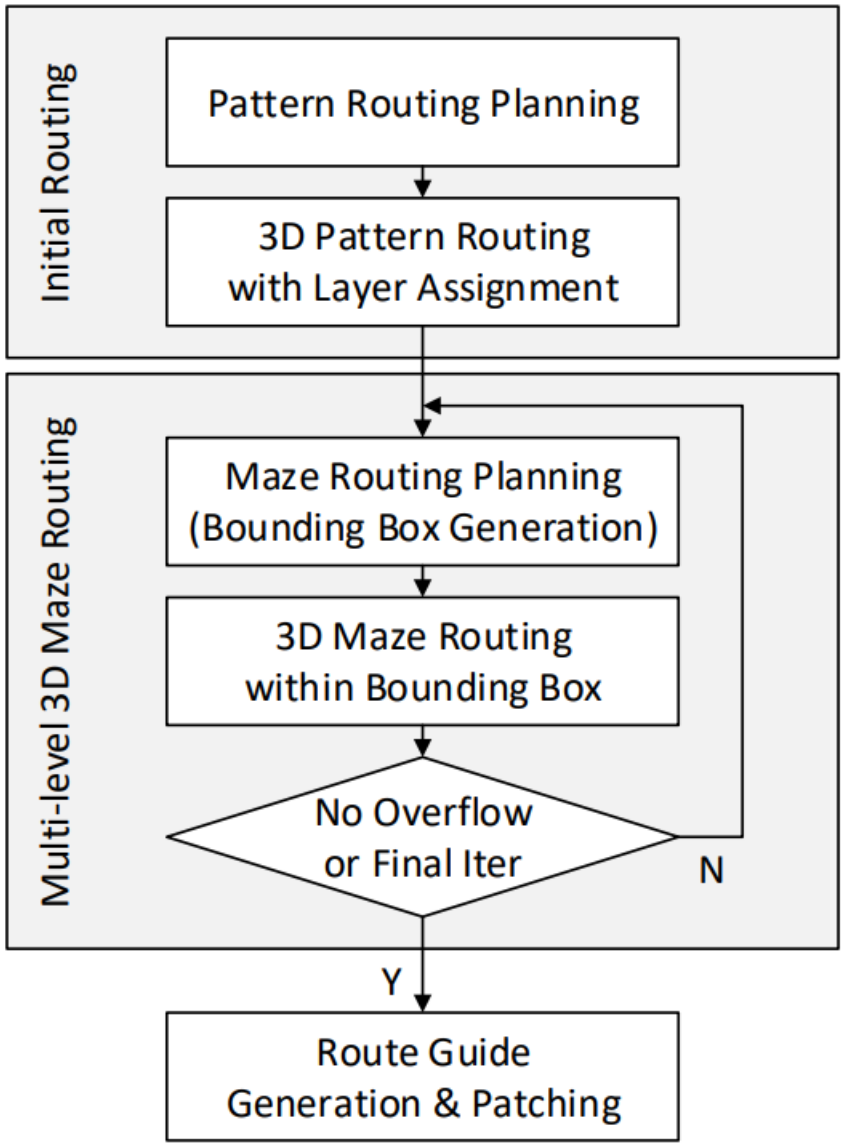


Fig. 1: Overall Flow of the Proposed Algorithm.

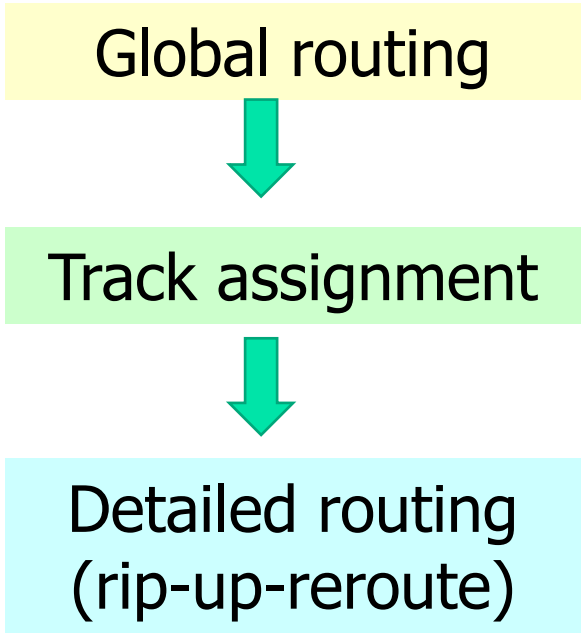
Other Global Routing Graphs

5.5 The Global Routing Flow – General Idea

1. Defining the routing regions (Region definition)
   - Layout area is divided into routing regions
   - Nets can also be routed over standard cells
   - Regions, capacities and connections are represented by a graph
2. Mapping nets to the routing regions (Region assignment)
   - Each net of the design is assigned to one or several routing regions to connect all of its pins
   - Routing capacity, timing and congestion affect mapping
3. Assigning crosspoints along the edges of the routing regions (Midway routing)
   - Routes are assigned to fixed locations or crosspoints along the edges of the routing regions
   - Enables scaling of global and detailed routing

After Global Routing

- Follow GR's guide, we start to do detailed routing
- Need to talk about pin access



Another Critical Problem – Pin Access

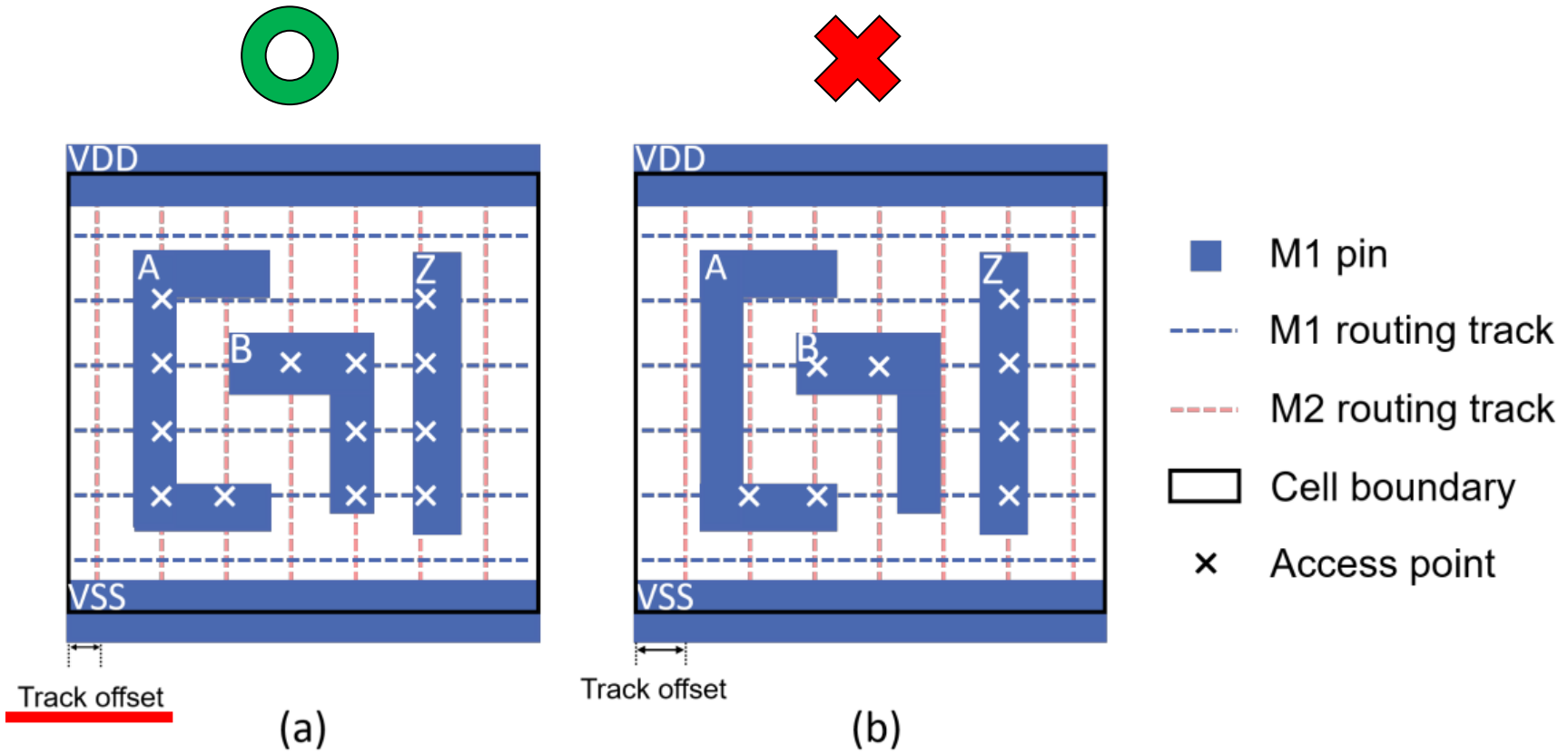
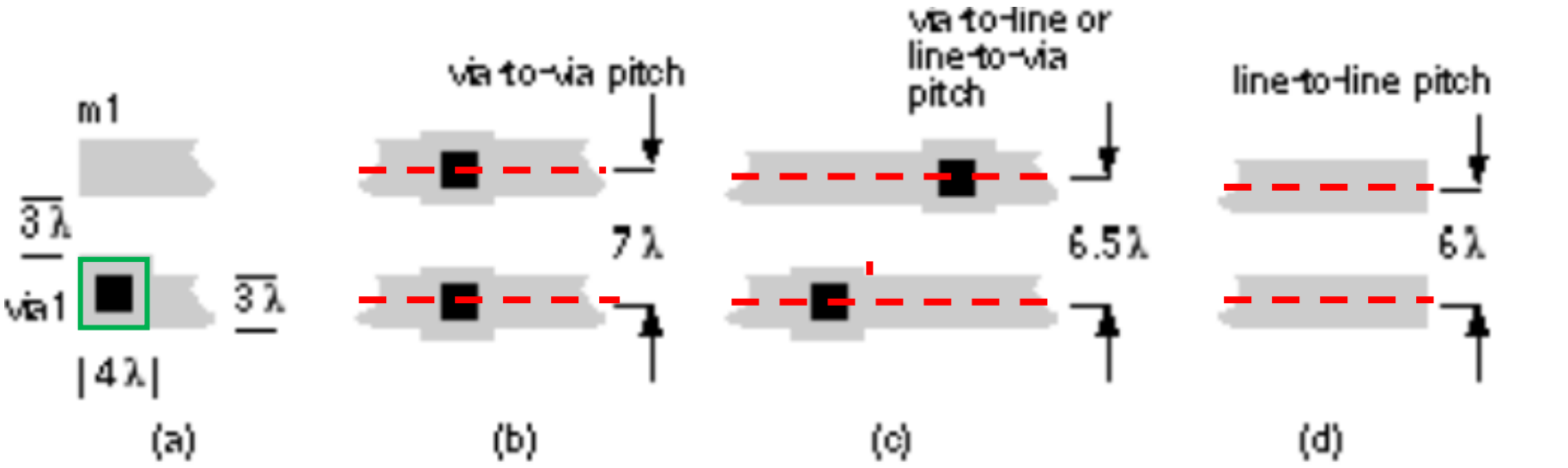


Fig. 1: Pin access points with different routing track-placement site offsets.

Routing Models

- **Grid-based model:**
  - A grid is super-imposed on the routing region.
  - Wires follow paths along the grid lines.
  - Pitch: distance between two grid lines (usually wire width +spacing)
    - But if contact is bigger, pitch likely will be

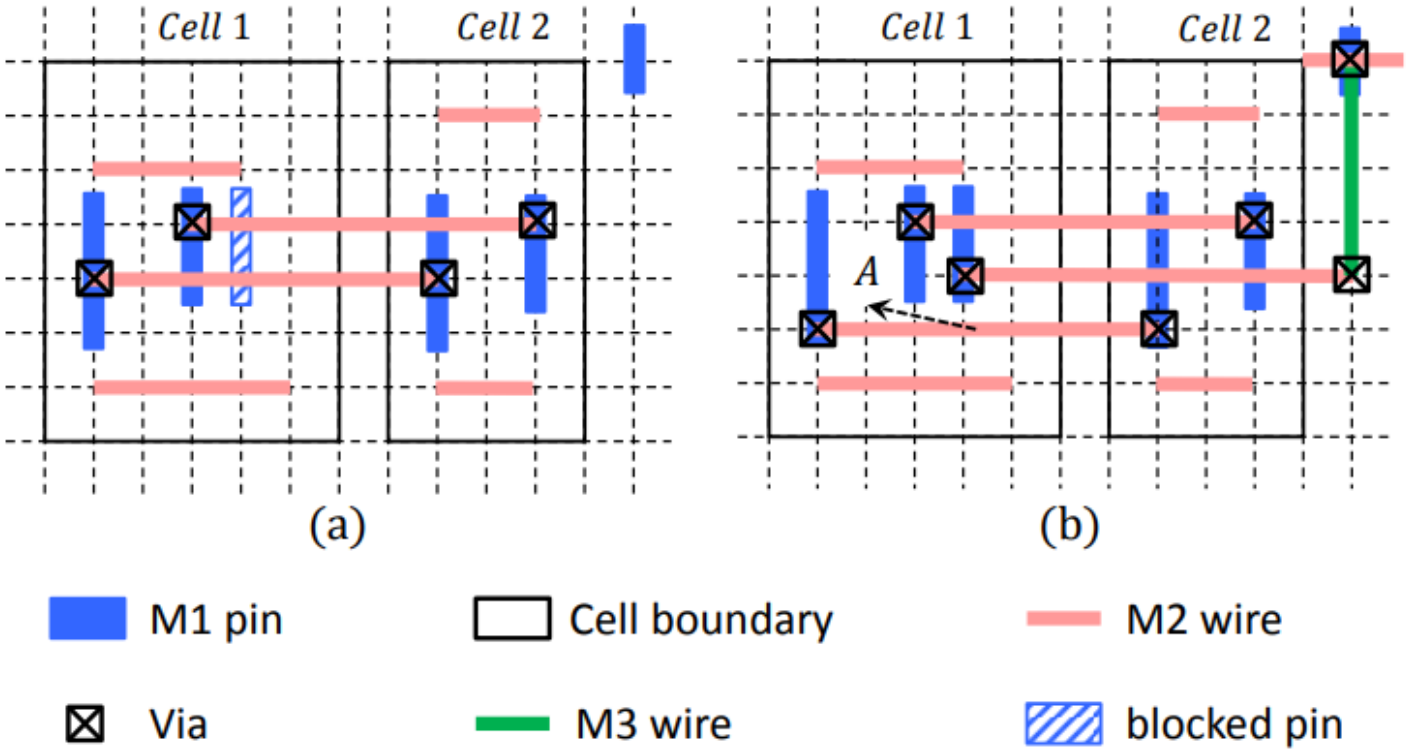


2(half contact width) + spacing

Line-to-line + ½(contact width – line width)

2(half width) + spacing

Two wires blocking "shaded blue pin"



One step in Rip-Up-Reroute is to see if a pin's access points were blocked, then remove neighboring nets

Figure 1: Pin access for detailed routing, (a) pin access failure, (b) pin access success.

HPC: hit point combination

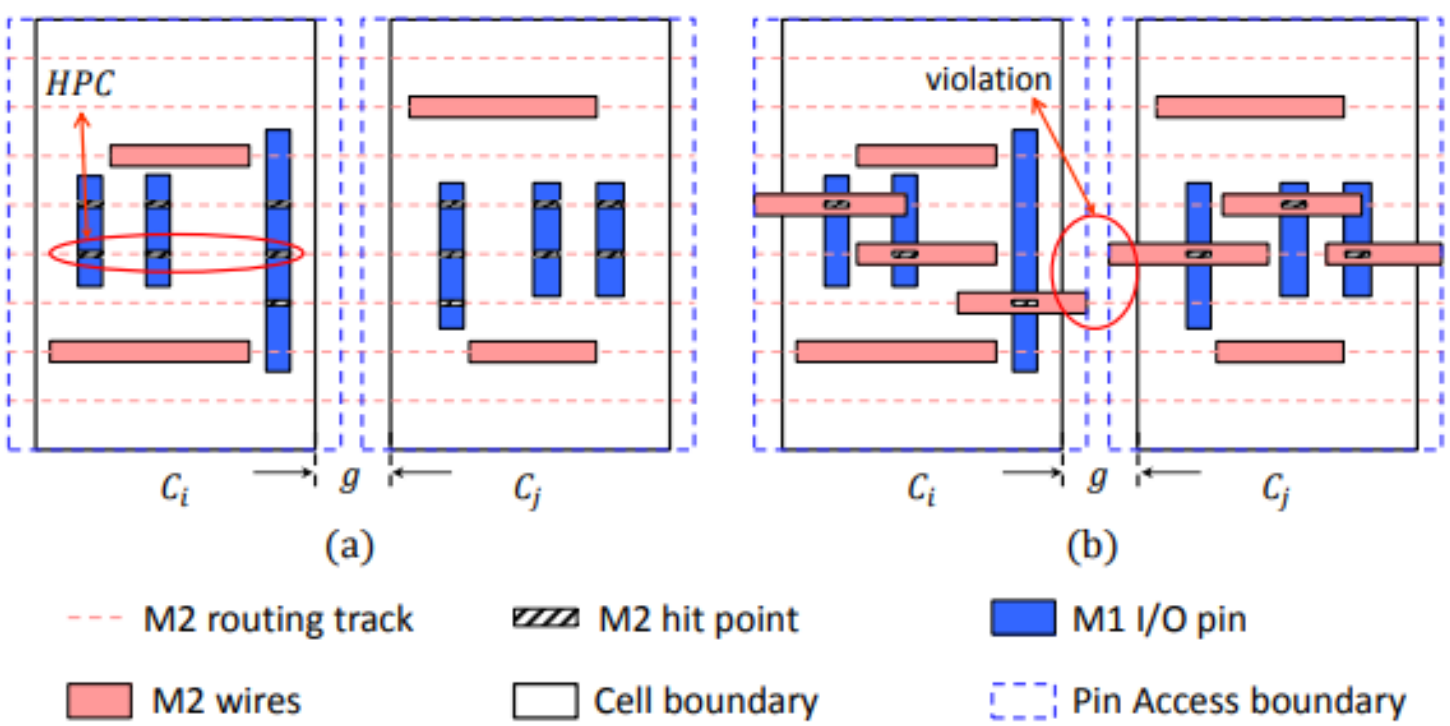


Figure 2: The intra-cell and inter-cell pin access, (a) M2 tracks and cell layout, (b) potential inter-cell pin access conflicts.

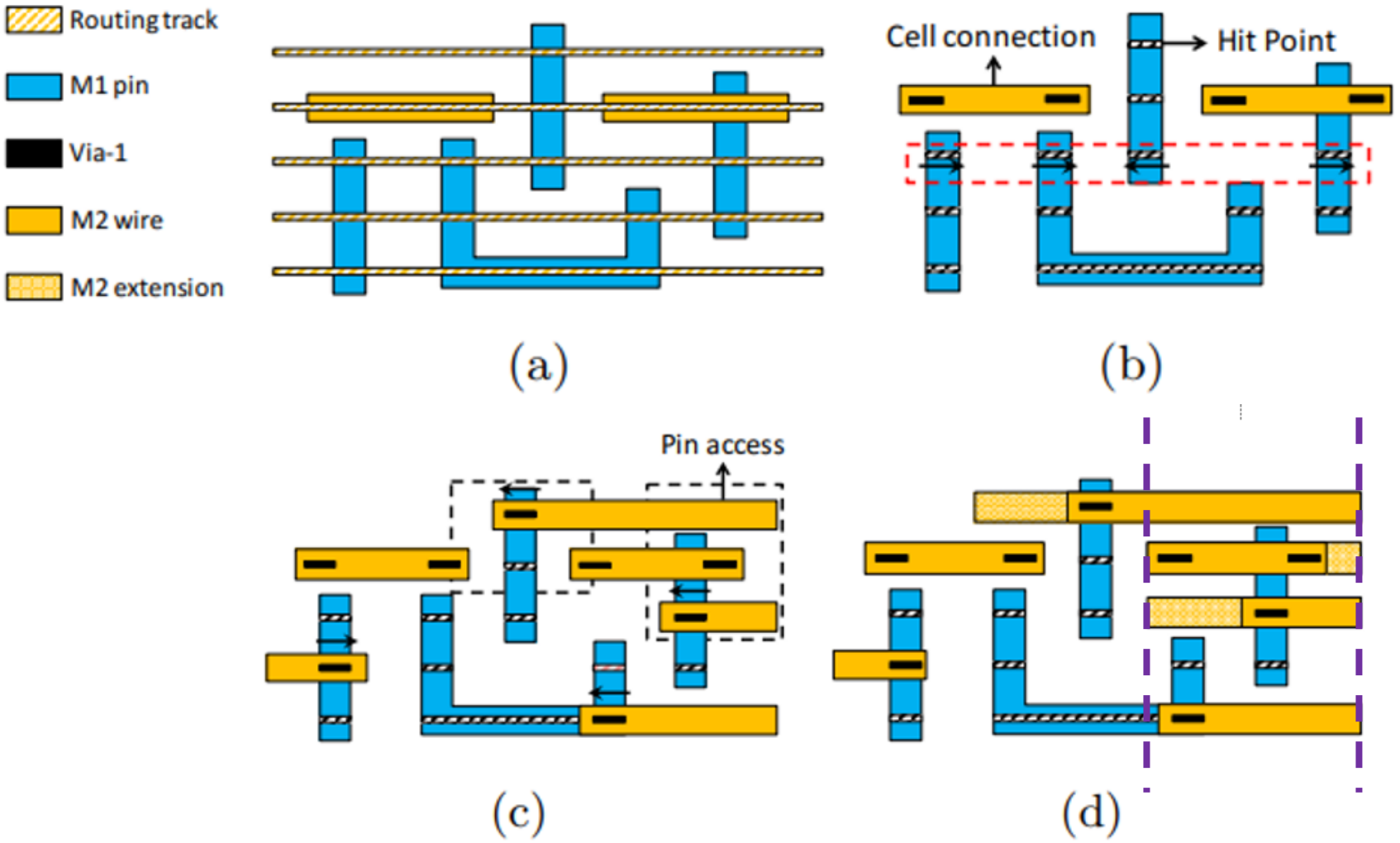


Figure 5. Standard pin access optimization, (a) standard cell I/O pins and M2 routing tracks, (b) hit points and M2 within-cell connections, (c) a hit point combination with M2 pin access, (d) SADP-friendly pin access with M2 extensions.[4]

Standard Cell Routability

standard_cell_routability.pdf (SPIE, 2020)

An Automated System for Checking Process Friendliness and Routability of Standard Cells

https://dl.acm.org/doi/pdf/10.1145/3569052.3571875?casa_token=y-0eF5f3r2YAAAAA:-rXGfkLKKJZ1H5obW0Nmd0gE0WBwhY4ItMA-uU1JDV5iXX7vH5rAgICQA2NfS2Lnyo2_YMvDP1Hzig

Pin Access-Oriented Concurrent Detailed Routing

https://dl.acm.org/doi/pdf/10.1145/3658617.3697731

Pin Access-aware Multiple Via Pillar Co-Design for Routability Optimization

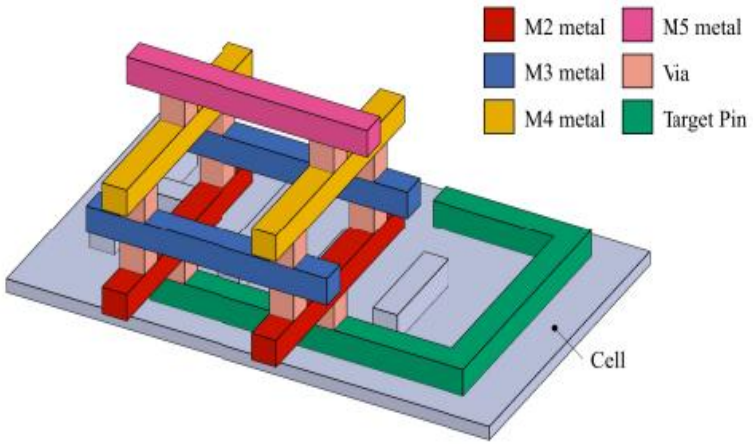


Figure 1: An illustration of the via pillar structure, which consists of parallel metal wires and multiple vias.

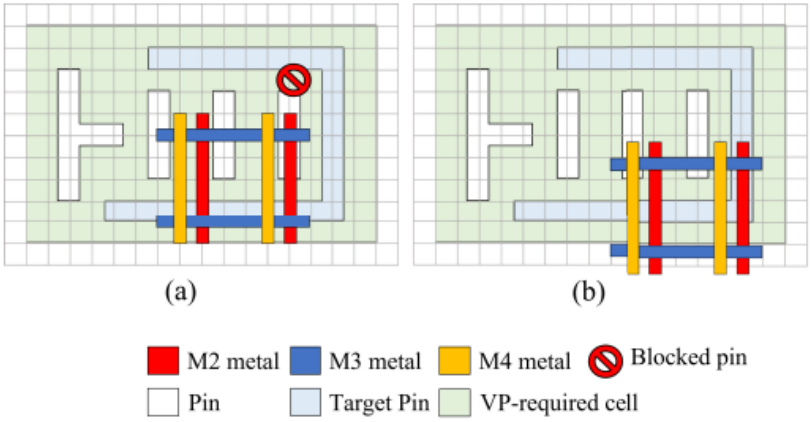


Figure 2: Via pillar design considering pin accessibility. (a) The construction of the via pillar blocks one of the pins of its cell. (b) Another via pillar structure keeps all pins unblocked.

https://dl.acm.org/doi/pdf/10.1145/3649329.3655918

Concurrent Detailed Routing with Pin Pattern Re-generation for Ultimate Pin Access Optimization

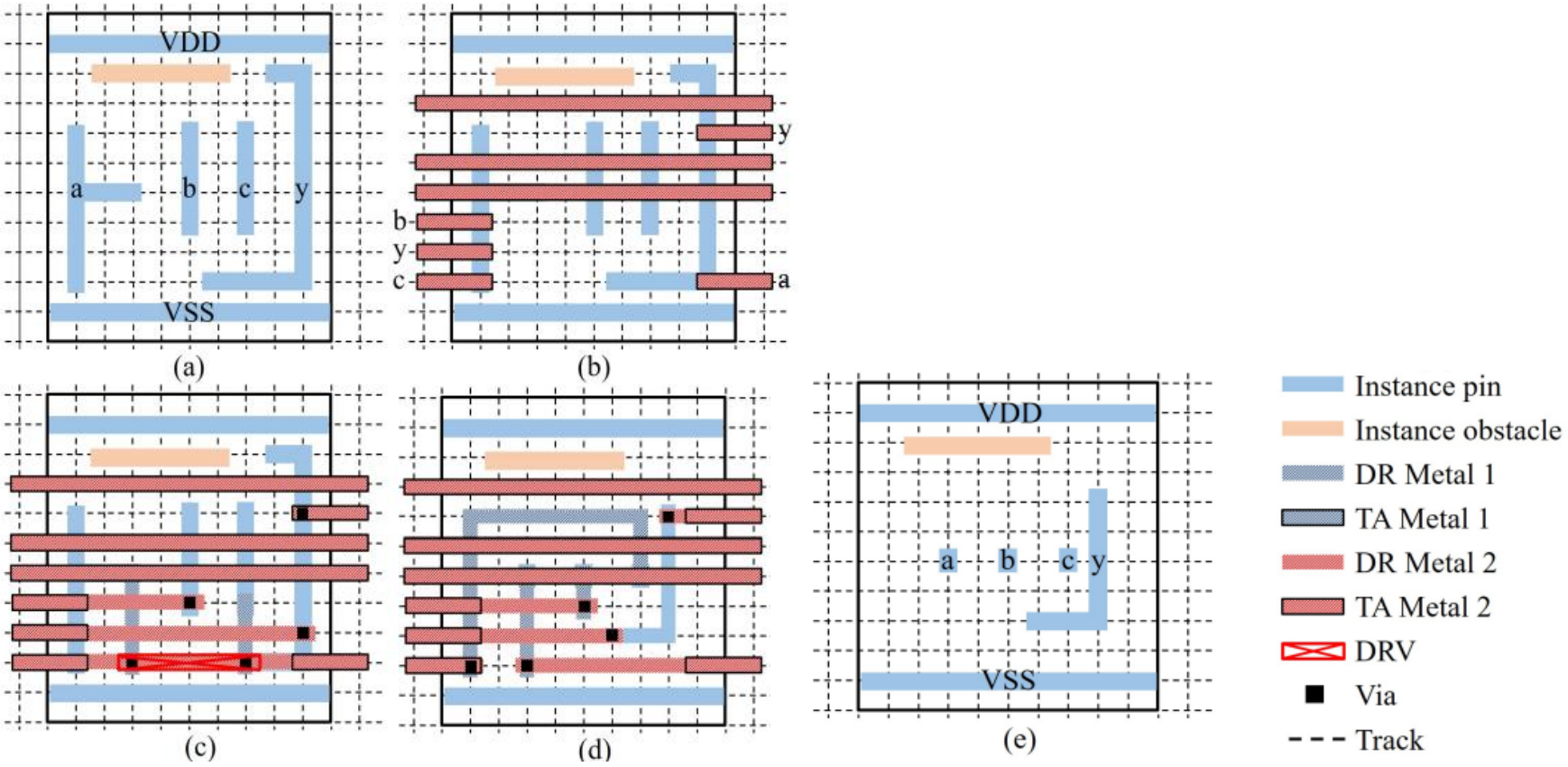


Figure 1: An example of concurrent detailed routing (DR) with pin pattern re-generation. (a) An instance with four pins *a*, *b*, *c*, and *y* from left to right. (b) An instance after track assignment (TA). (c) A detailed routing result after the original pin pattern. (d) A detailed routing result after the proposed flow. (e) The re-generated pin pattern after the proposed flow.

Detailed Routing

- Track assignment using GR's guides
- Router design rule checker
- Area based rip-up and re-route

Detailed Routing: Rip-Up and Re-routing

- Rip-up and re-routing is required if a global or detailed router fails in routing some nets.
- Approaches: the manual approach? the automatic procedure?
- Two steps in rip-up and re-routing
  1. Identify bottleneck regions, rip off some already routed nets.
  2. Route the blocked connections, and re-route the ripped-up connections
  3. But need to avoid falling into seesaw
- Repeat the above steps until all connections are routed or a time limit is exceeded.

- Difficult, but very important & powerful

How to detect DRC errors during routing?

Latest Attempt In School For Detailed Routing

https://github.com/cuhk-eda/dr-cu

Dr. CU

Dr. CU is a VLSI detailed routing tool developed by the research team supervised by Prof. Evangeline F.Y. Young in The Chinese University of Hong Kong (CUHK). Different from global routing, detailed routing takes care of many detailed design rules and is performed on a significantly larger routing grid graph. In advanced technology nodes, it becomes the most complicated and time-consuming stage in the VLSI physical design flow. To tackle the challenges, we design and implement several efficient and effective data structures and algorithms under a holistic framework:

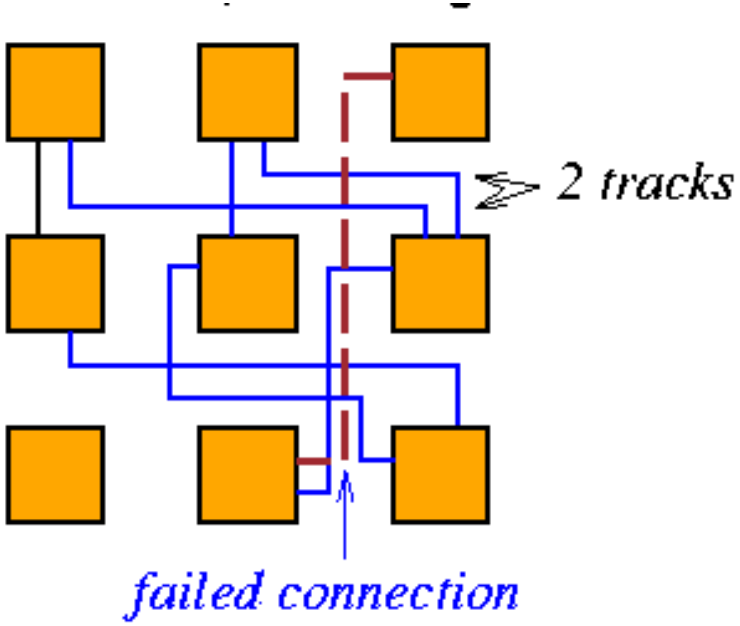
- A set of two-level sparse data structures
- An optimal correct-by-construction path search
- An efficient bulk synchronous parallel scheme
- ...

Course Ordering Of The Major Topics

- Digital design flow
- CMOS logic gates and Boolean equations
- Algorithms and complexity
- Compaction
- Partitioning
- Floorplanning
- Placement
- Basic logic synthesis, technology mapping
- High level synthesis
- Routing
- Clock/PG routing & Elmore delay & Clock tree synthesis
- Simulation

Global Routing in Gate Array

- Objective
  - **Guarantee 100% routability.**
- For high performance,
  - Minimize the maximum wire length.
  - Minimize the maximum path length.

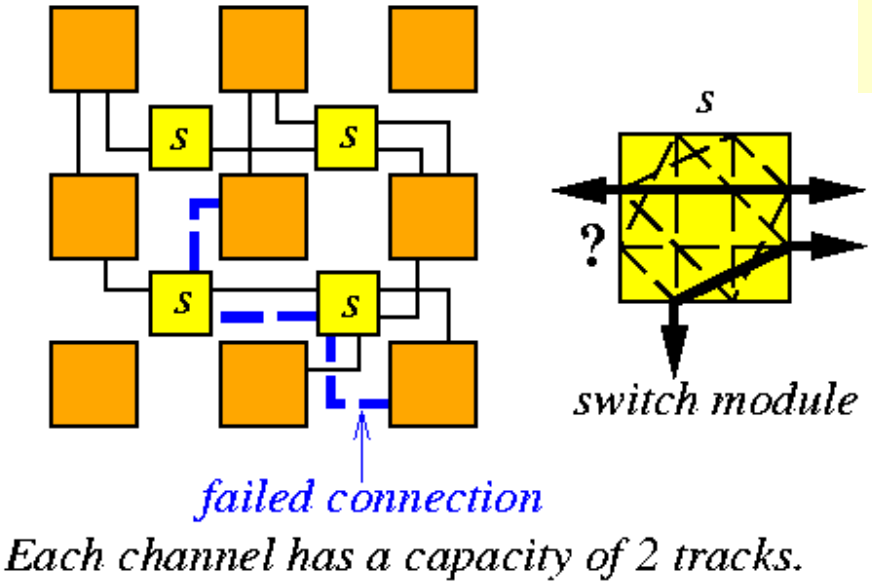


Each channel has a capacity of 2 tracks.

Global Routing in FPGA

- Objective
  - Guarantee 100% routability.
  - Consider **switch-module architectural constraints.**
- For performance-driven routing,
  - **Minimize # of switches used.**
  - Minimize the maximum wire length.
  - Minimize the maximum path length.

There are global interconnect lines. Routing is very specific.



Each channel has a capacity of 2 tracks.