Logistics

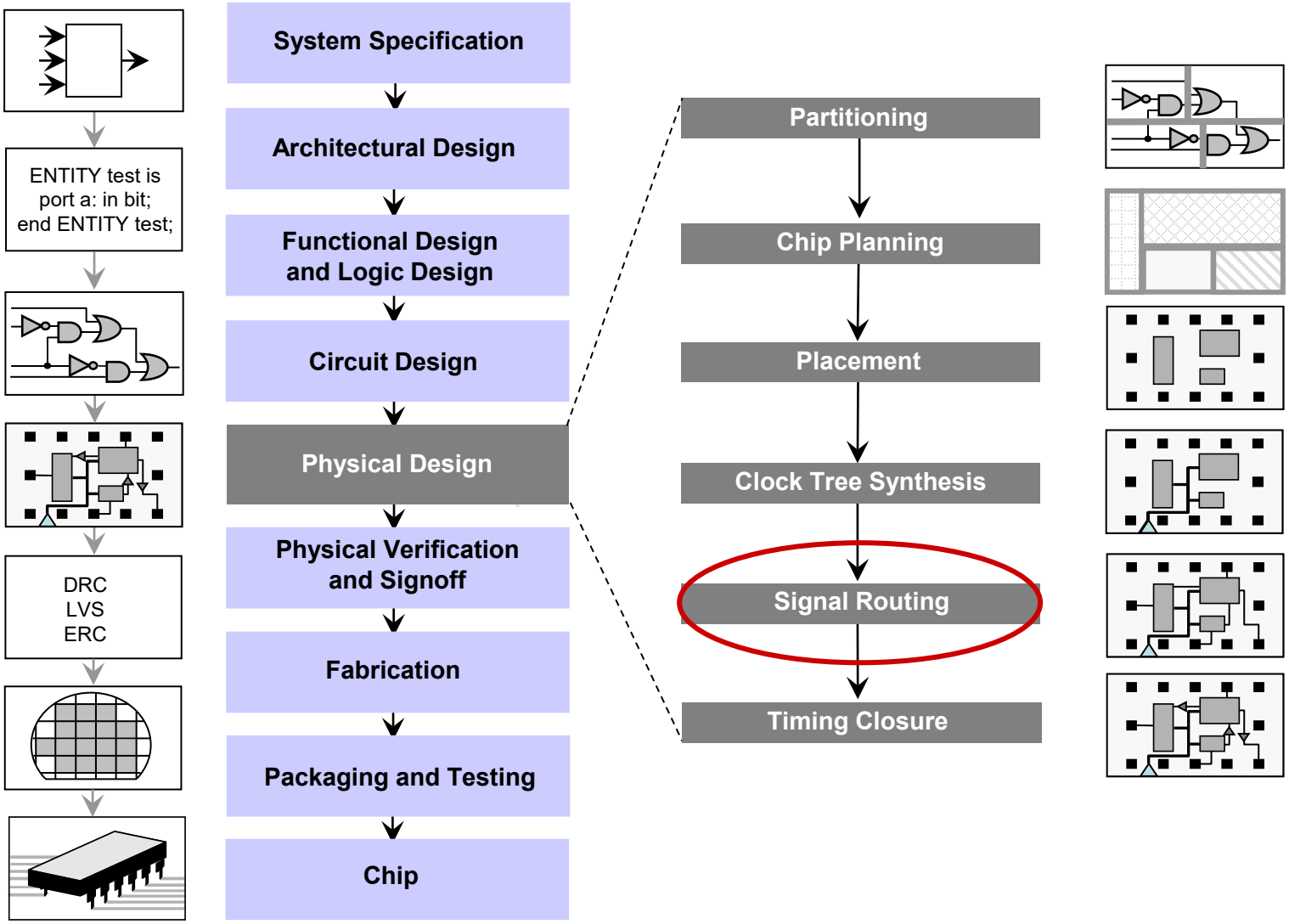
- Work on your project!
  - Midterm project report is **1 slide** uploaded on Gradescope.
- No lab this week

Lecture 10: Global Routing

ECE201A

Some notes adopted from
Andrew B. Kahng
Lei He
Igor Markov
Mani Srivastava
Mohammad Tehranipoor

Introduction



Introduction: General Routing Problem

Netlist:

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

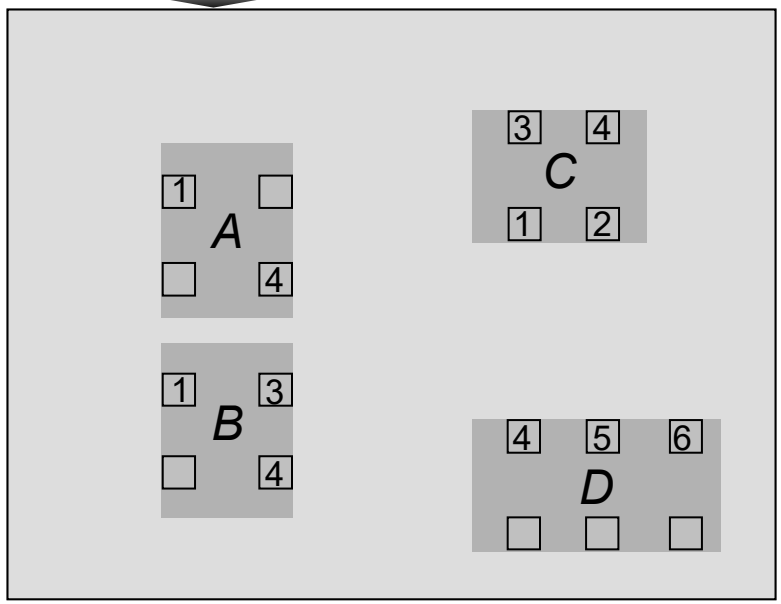
$N_2 = \{D_4, B_4, C_1, A_4\}$

$N_3 = \{C_2, D_5\}$

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

Technology Information (Design Rules)

Placement result



Introduction: General Routing Problem

Netlist:

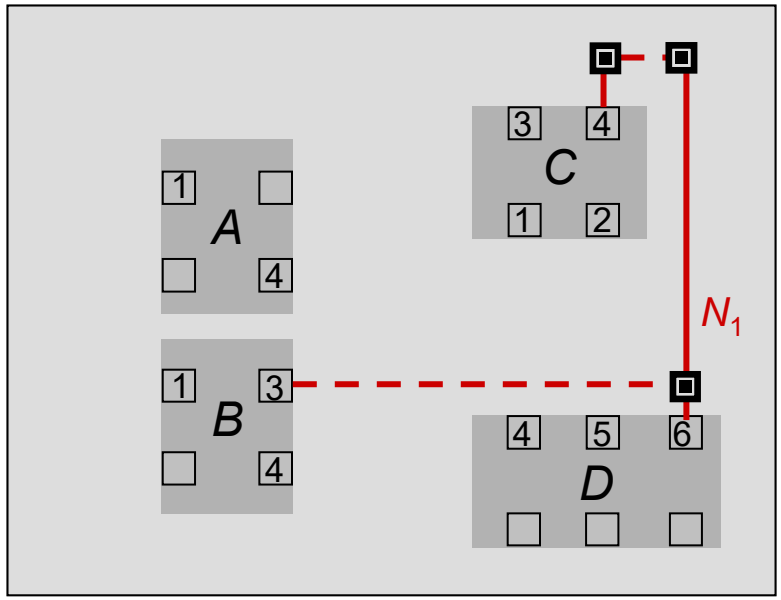
$N_1 = \{C_4, D_6, B_3\}$

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

$N_3 = \{C_2, D_5\}$

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

Technology Information (Design Rules)



Introduction: General Routing Problem

Netlist:

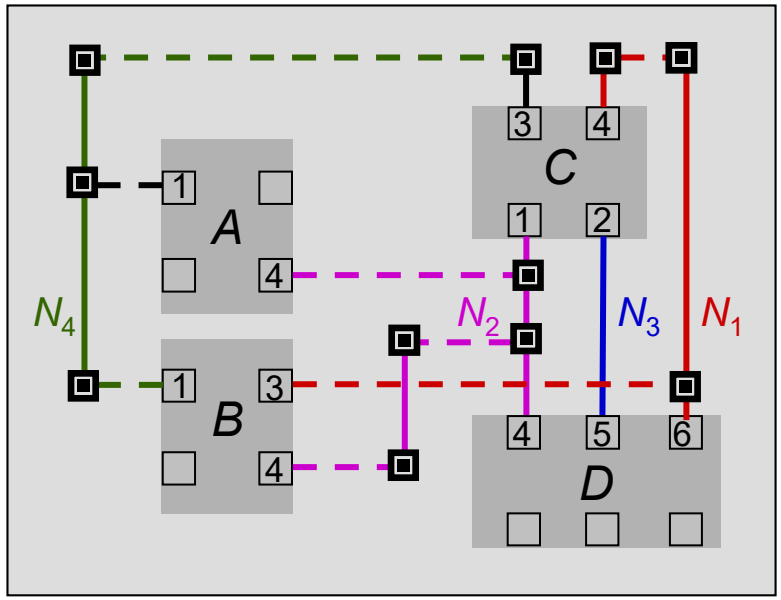
$N_1 = \{C_4, D_6, B_3\}$

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

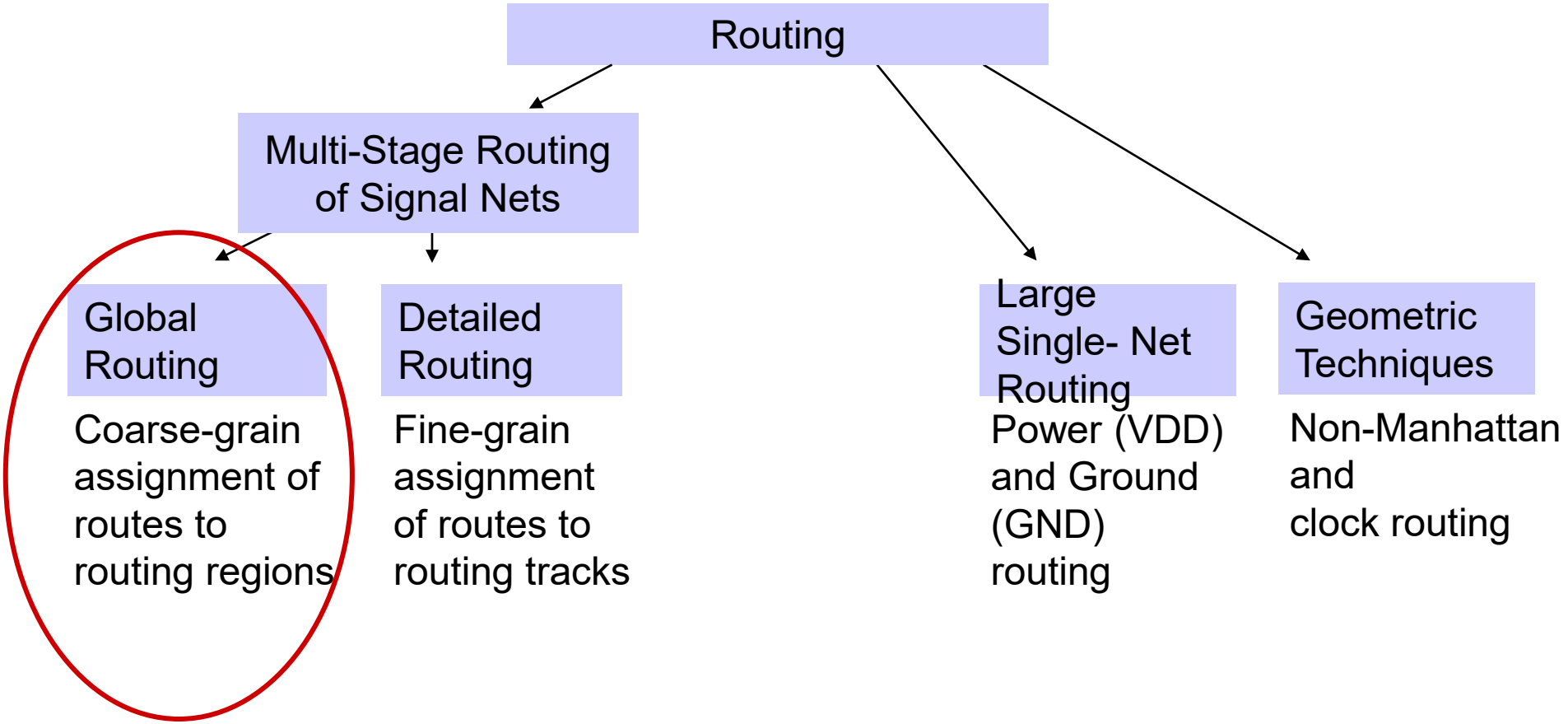
$N_3 = \{C_2, D_5\}$

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

Technology Information (Design Rules)



Taxonomy of Routing

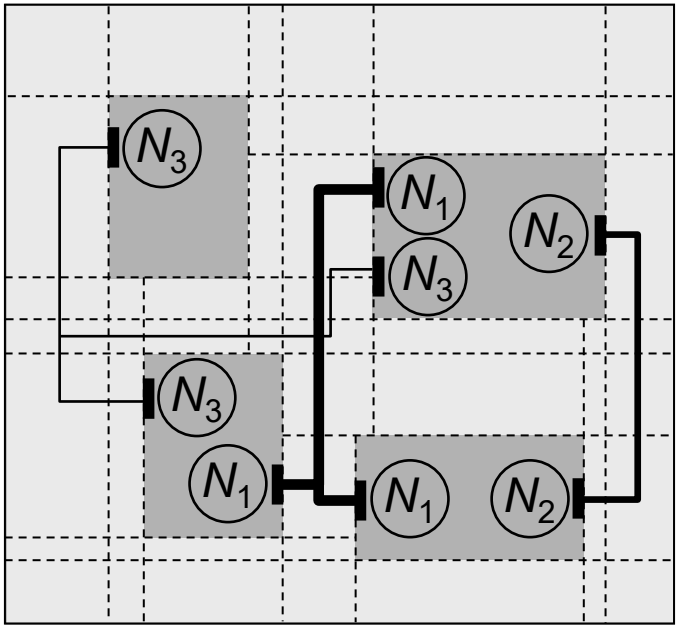


Global Routing Introduction

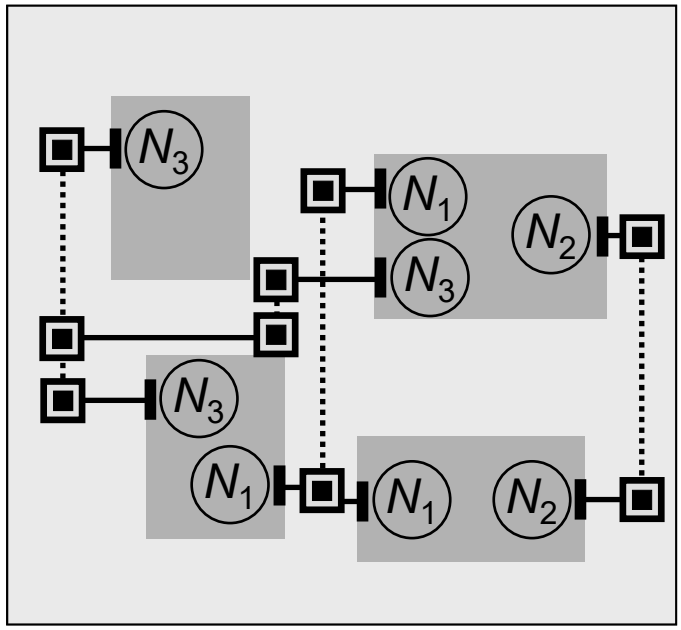
- Wire segments are tentatively assigned (embedded) within the chip layout
  - determine whether a given placement is routable, and
  - determine a coarse routing for all nets within available routing regions
- Chip area is represented by a coarse routing grid
- Available routing resources are represented by edges with capacities in a grid graph
  - Nets are assigned to these routing resources
- Considers goals such as
  - minimizing total wirelength, and
  - reducing signal delays on critical nets

Global vs. Detailed

Global Routing



Detailed Routing

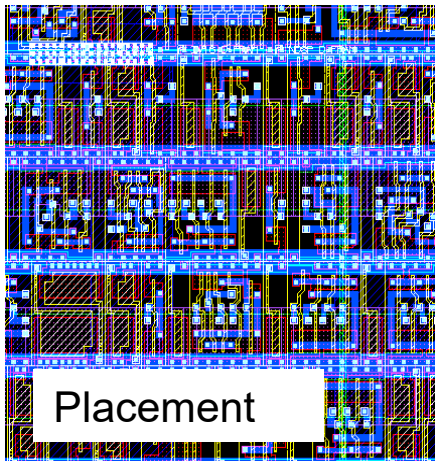


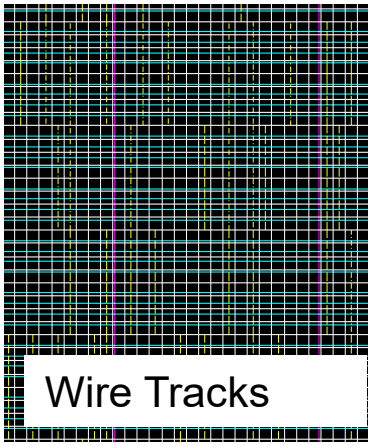
Horizontal Segment

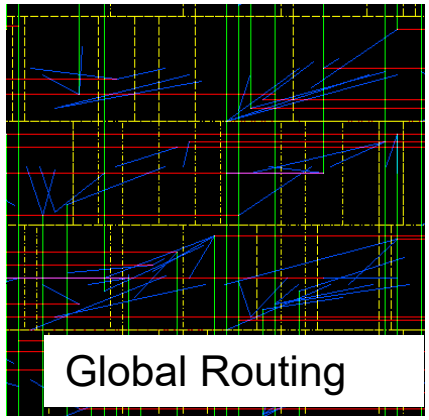
Vertical Segment

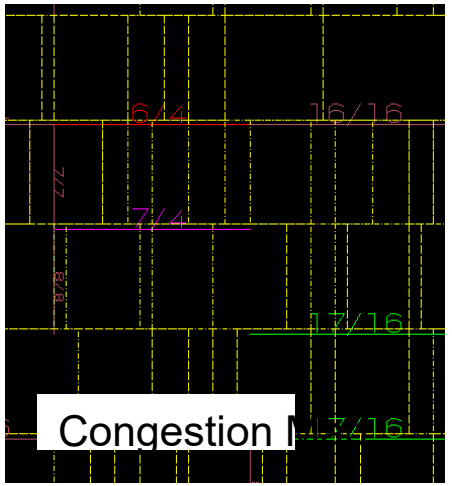
Via

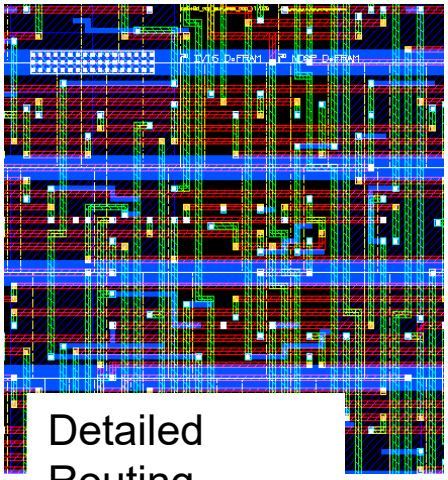
Global vs. Detail











Before we move on..

- A LEF example

```
LAYER M2
  TYPE ROUTING ;
  WIDTH 0.07 ;
  SPACING 0.07 ;
  PITCH 0.19 ;
  DIRECTION VERTICAL ;
  RESISTANCE RPERSQ 0.25 ;
  THICKNESS 0.14 ;
  CAPACITANCE CPERSQDIST 0.001 ;
END M2
LAYER V2
  TYPE CUT ;
  SPACING 0.075 ;
  WIDTH 0.065 ;
END V2
MACRO NOR2_X1
  CLASS core ;
  ORIGIN 0 0 ;
  SYMMETRY X Y ;
  SITE Free_OMC_Si2_PDK45nm ;
  SIZE 0.57 BY 1.4 ;
END VDD
  PIN A1
    DIRECTION INPUT ;
    PORT
      LAYER M1 ;
        POLYGON 0.37 0.58 0.505 0.58 0.505 0.68 0.37 0.68  ;
    END
```

Layer definitions

Cell definition

Pin definition

Compare with GDS → Do for a Nangate cell

Before we move on..

- A DEF example

```
UNITS DISTANCE MICRONS 2000 ;
DIEAREA ( 0 0 ) ( 600970 600160 ) ;
ROW CORE_ROW_0 TSM90NMSITE 20720 575120 FS DO 1000 BY 1 STEP 560 0 ;
ROW CORE_ROW_1 TSM90NMSITE 20720 570080 N DO 1000 BY 1 STEP 560 0 ;
ROW CORE_ROW_2 TSM90NMSITE 20720 565040 FS DO 1000 BY 1 STEP 560 0 ;
ROW CORE_ROW_3 TSM90NMSITE 20720 560000 N DO 1000 BY 1 STEP 560 0 ;
ROW CORE_ROW_4 TSM90NMSITE 20720 554960 FS DO 1000 BY 1 STEP 560 0 ;
ROW CORE_ROW_5 TSM90NMSITE 20720 549920 N DO 1000 BY 1 STEP 560 0 ;
ROW CORE_ROW_6 TSM90NMSITE 20720 544880 FS DO 1000 BY 1 STEP 560 0 ;
ROW CORE_ROW_7 TSM90NMSITE 20720 539840 N DO 1000 BY 1 STEP 560 0 ;
COMPONENTS 11553 ;
- us22/U426 AO22X2 + PLACED ( 174160 272720 ) S;
- us22/U427 AO2B2X2 + PLACED ( 162400 287840 ) N;
- us22/U428 OAI211XL + PLACED ( 228480 383600 ) FS;
- us22/U429 BUFX2 + PLACED ( 336560 413840 ) S;
NETS 11971 ;
- key[127]
  ( PIN key[127] ) ( u0/U746 B0 )
  + ROUTED M2 ( 88200 557480 ) ( * 560840 ) VIA2_X
    NEW M4 ( 88200 560840 ) ( * 580440 ) VIA3_X
    NEW M3 ( 19880 580440 ) ( 88200 * )
    NEW M2 ( 19880 572320 ) ( * 580440 ) VIA2_X
    NEW M3 ( 140 572320 0 ) ( 19880 * ) VIA2_X
    NEW M2 ( 88200 557480 ) VIA1_V
    NEW M4 ( 88200 560840 ) VIA3_TOS_E;
```

Database Unit (DBU)

Layout area in DBU

Cell Rows

Cells and placement

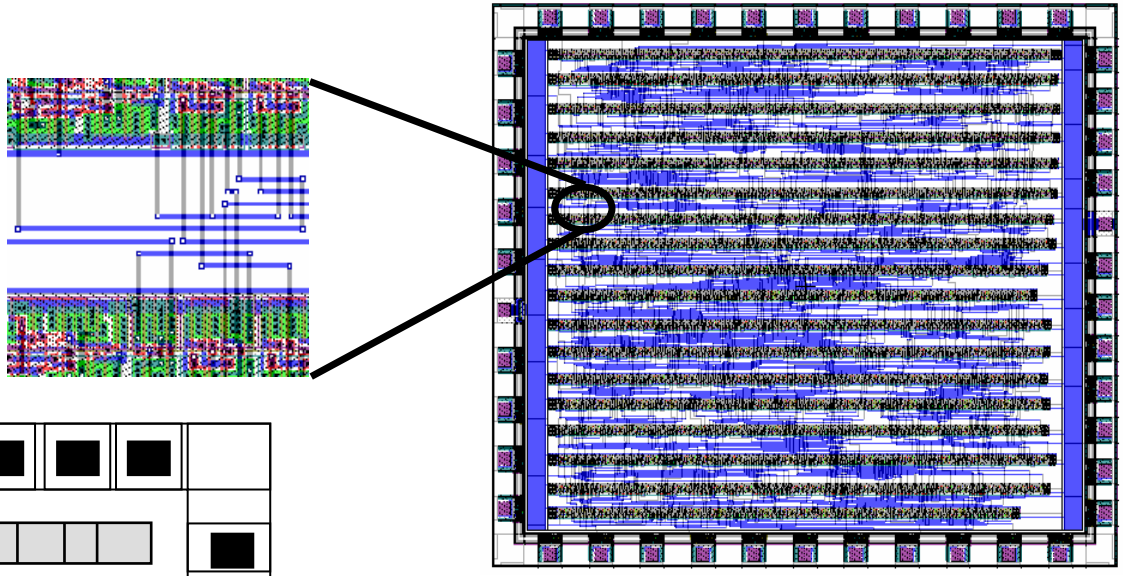
Nets and routing

Terminology and Definitions

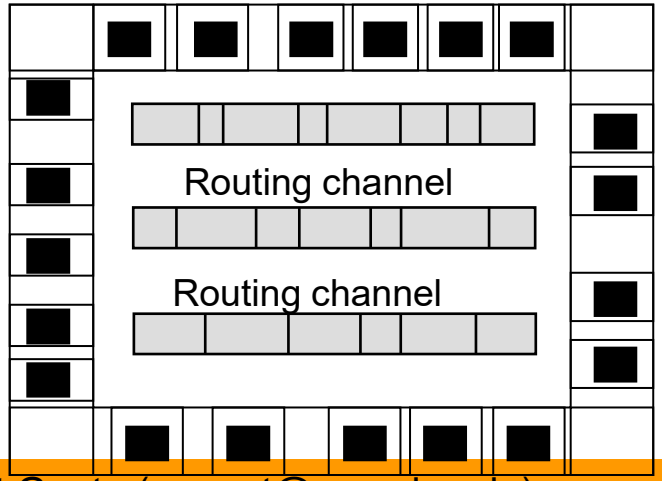
- Routing Track: Horizontal wiring path
- Routing Column: Vertical wiring path
- Routing Region: Region that contains routing tracks or columns
- Uniform Routing Region: Evenly spaced horizontal/vertical grid
- Non-uniform Routing Region: Horizontal and vertical boundaries that are aligned to external pin connections or macro-cell boundaries resulting in routing regions that have differing sizes

Routing Channel

Rectangular routing region with pins on two opposite sides



Standard cell layout (Two-layer routing)



Routing Capacity

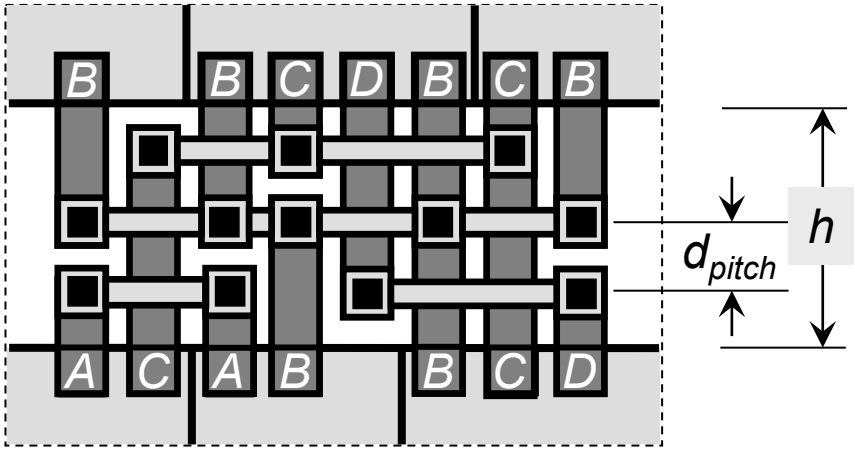
Number of available routing tracks or columns

- For single-layer routing, the capacity is the height h of the channel divided by the pitch d_{pitch}

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

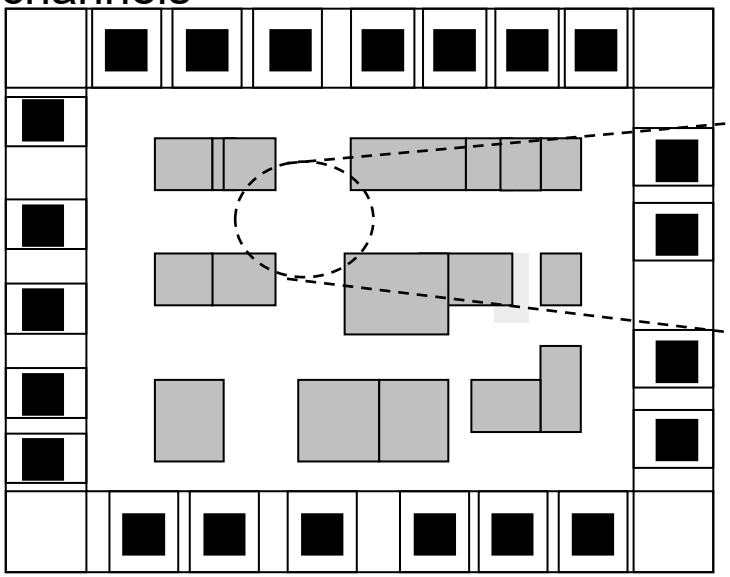
- For multilayer routing, the capacity σ is the sum of the capacities of all layers.

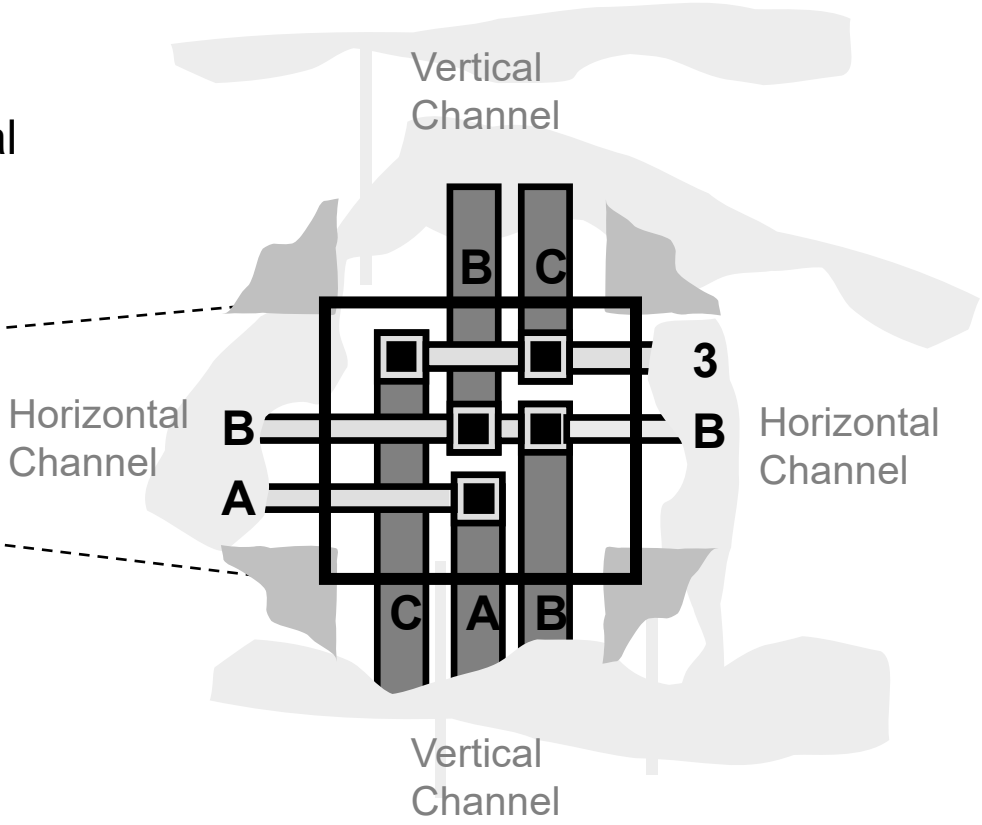
Horizontal Routing Channel



Switchbox

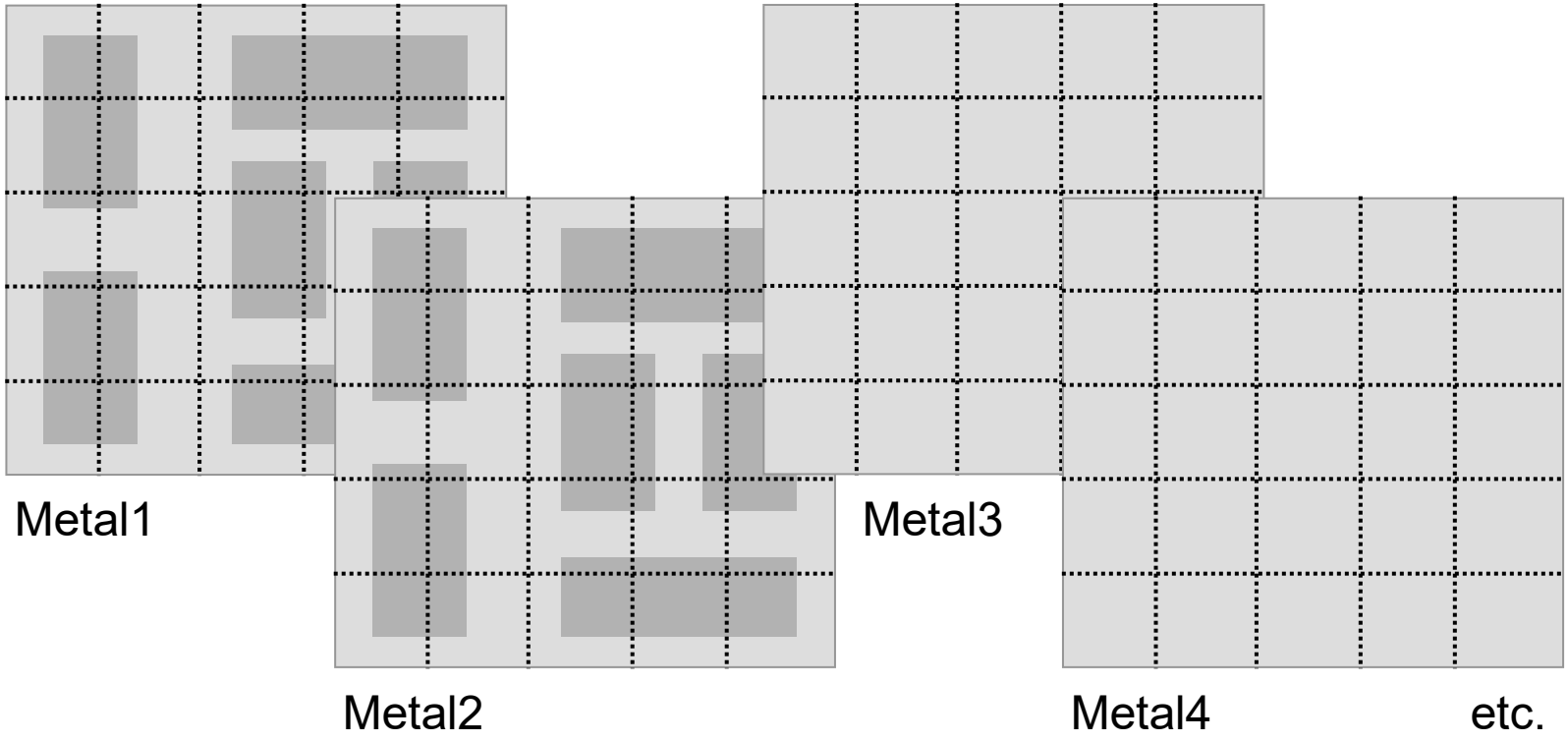
Intersection of horizontal and vertical channels



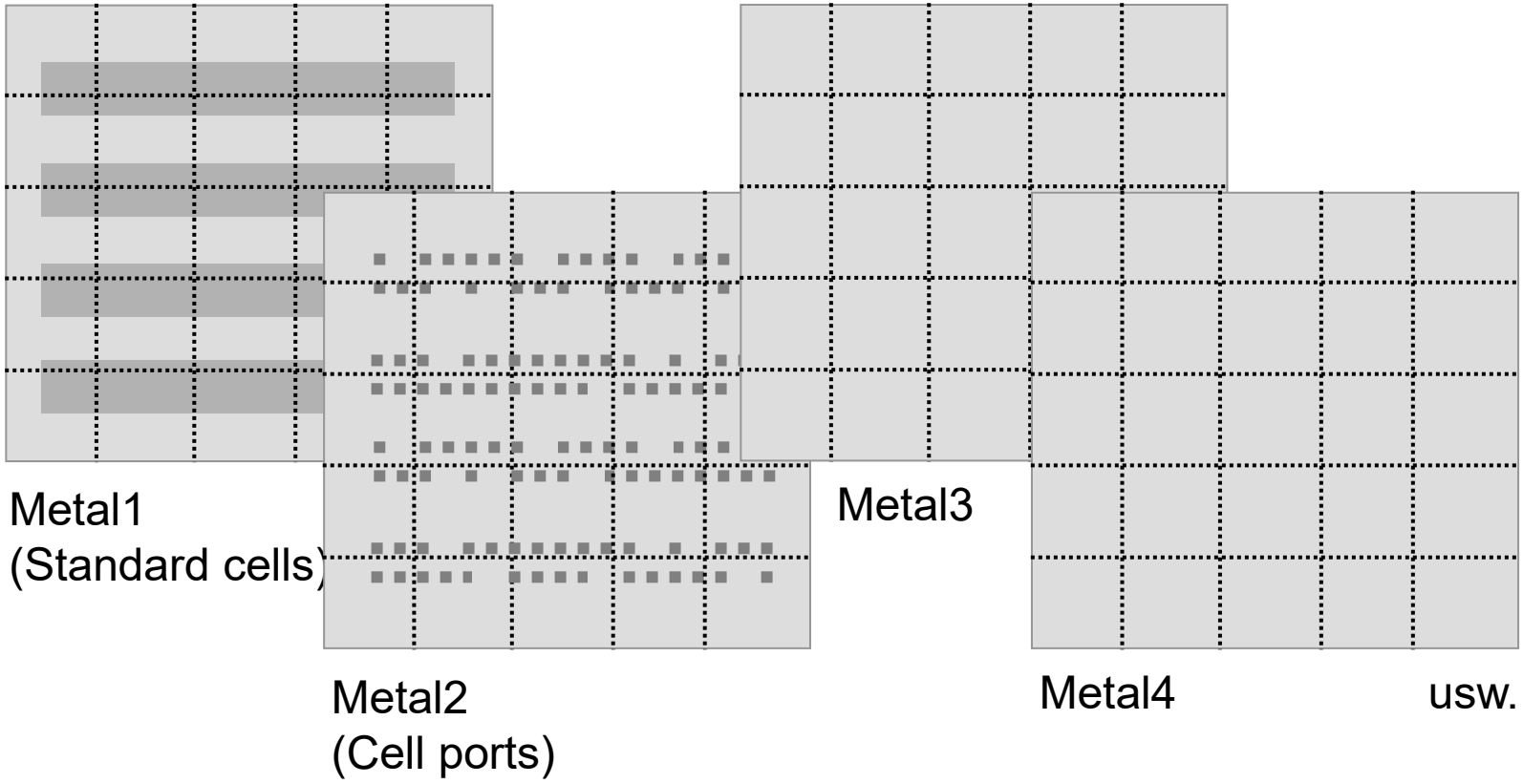


Horizontal channel is routed after vertical channel is routed

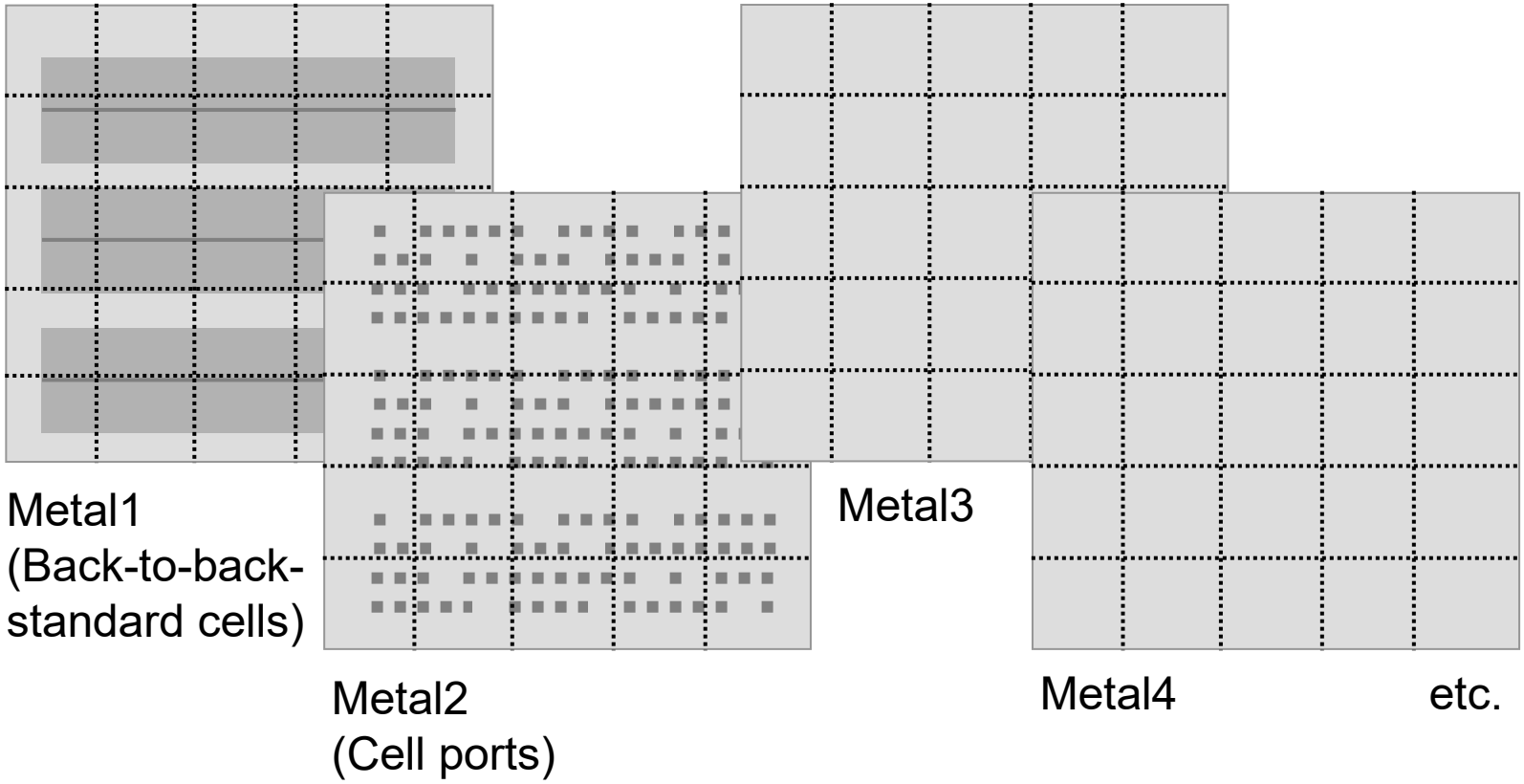
Gcells (tiles) with Macros



Gcells (Tiles) with standard cells



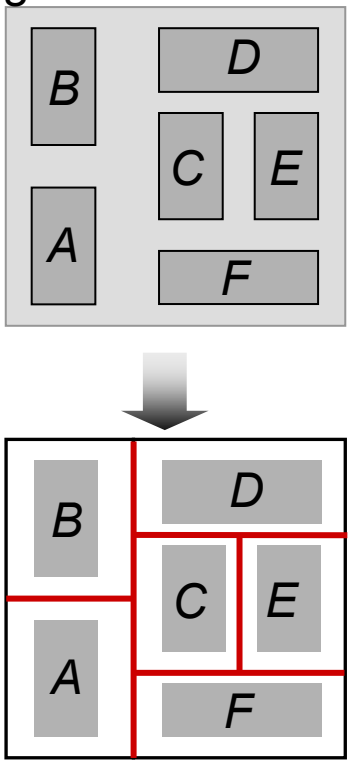
Gcells (Tiles) with standard cells (back-to-back)



Routing Contexts

Full-custom design

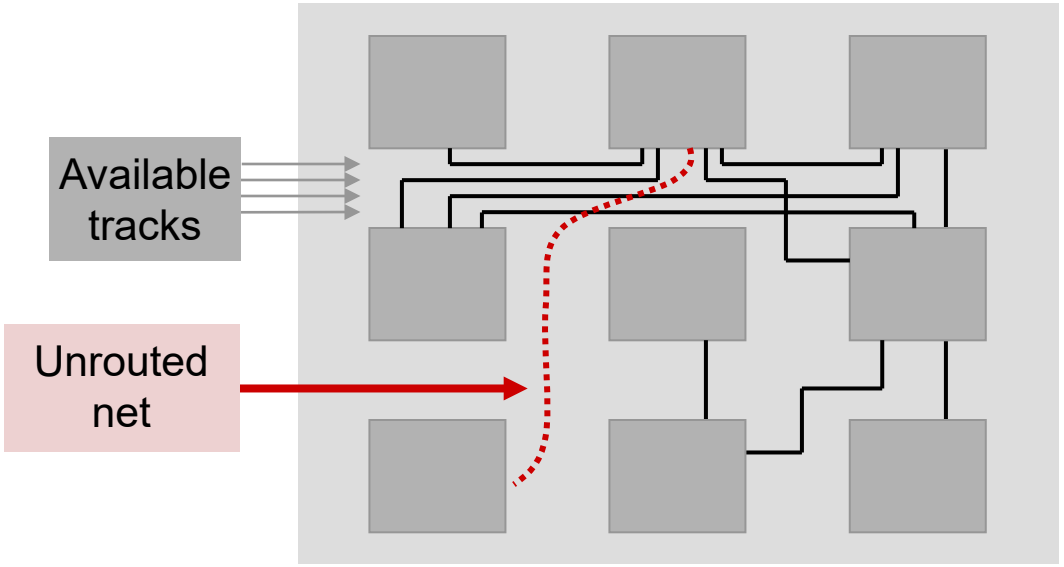
Layout is dominated by macro cells and routing regions are non-uniform



Types of channels

Gate-array design

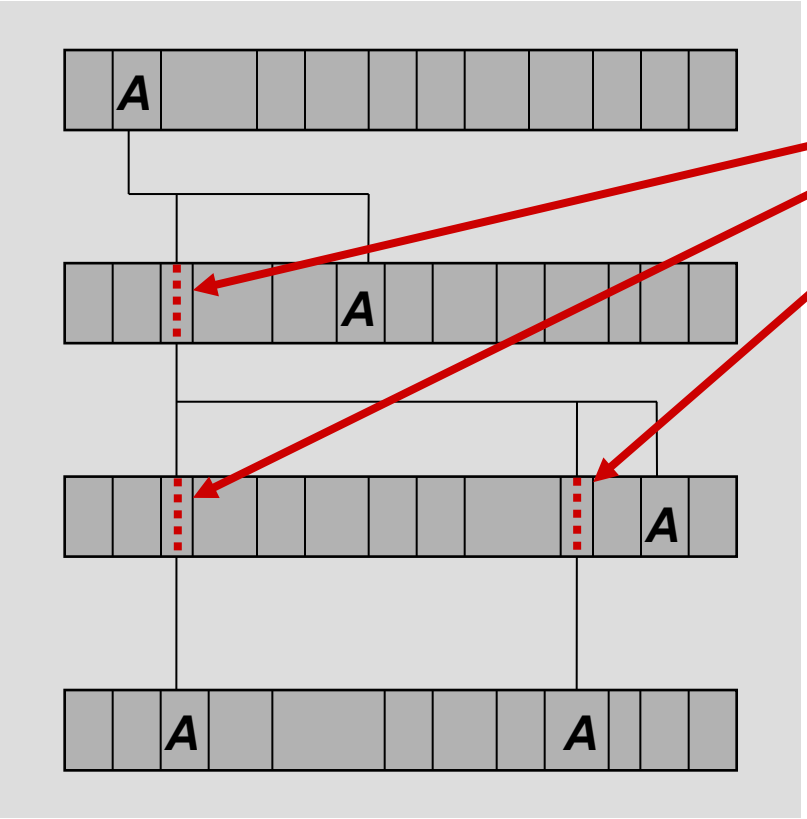
Cell sizes and sizes of routing regions between cells are fixed



Cell Routing Context

Standard-cell design

If number of metal layers is limited, feedthrough cells must be used to route across multiple cell rows



Feedthrough cells

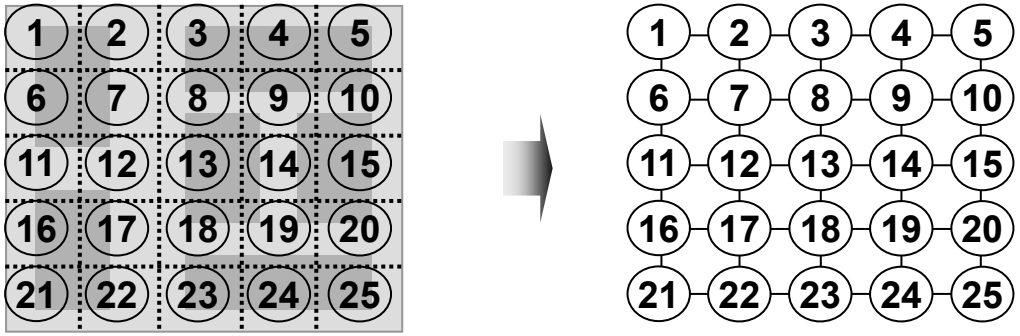
Variable-die, standard cell design:

Total height = ΣCell row heights + All channel heights

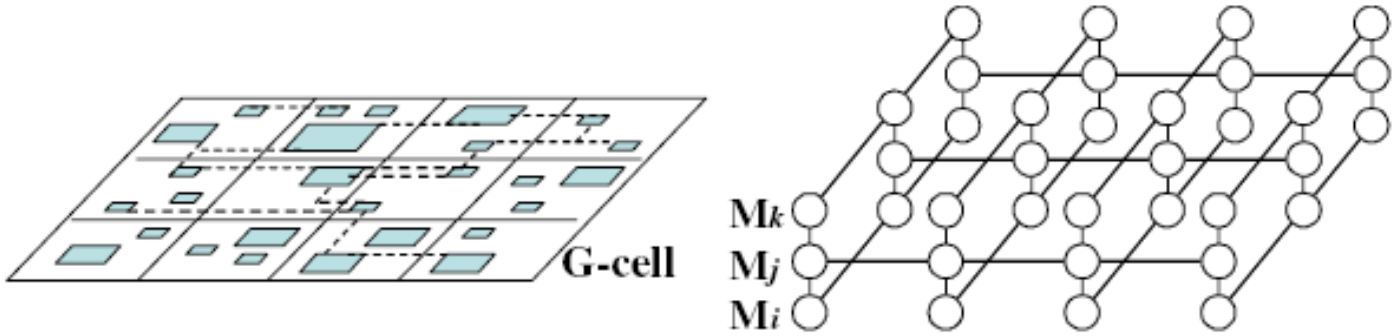
Representations of Routing Regions

- Routing regions are represented using efficient data structures
- Routing context is captured using a graph, where
  - nodes represent routing regions and
  - edges represent adjoining regions
- Capacities are associated with both edges and nodes
  to represent available routing resources

Grid Graph Model

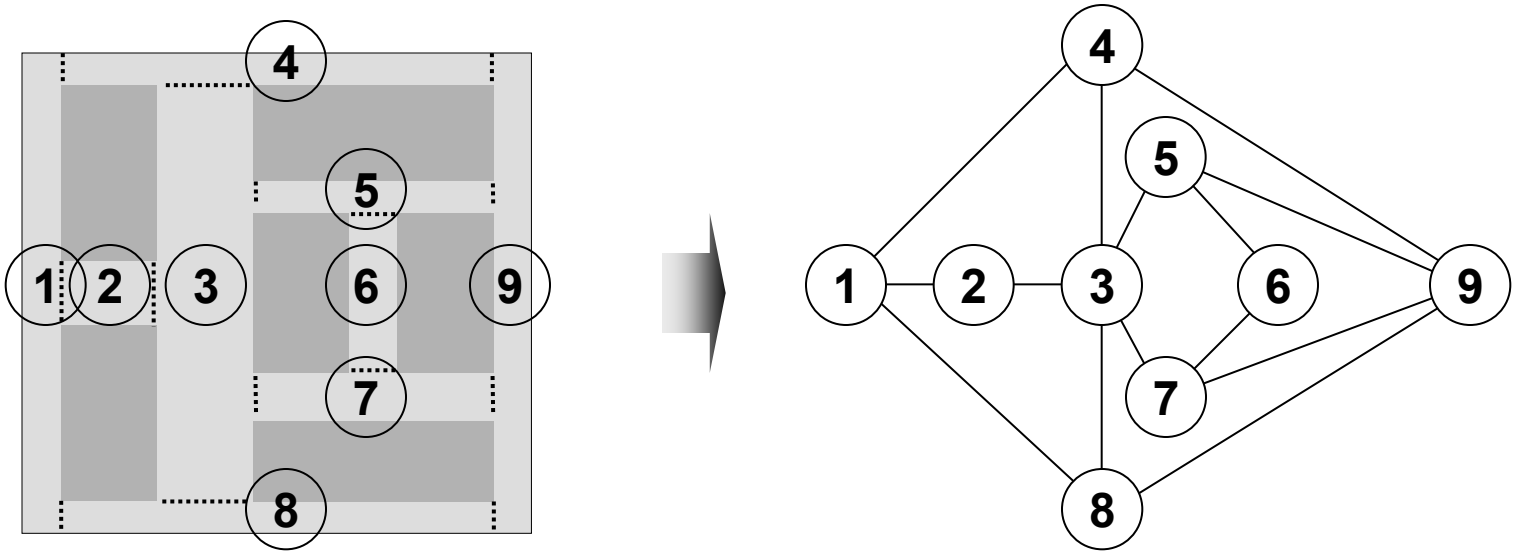


ggrid = (*V*,*E*), where the nodes $v \in V$ represent the routing grid cells (*gcells*) and the edges represent connections of grid cell pairs (v_i,v_j)



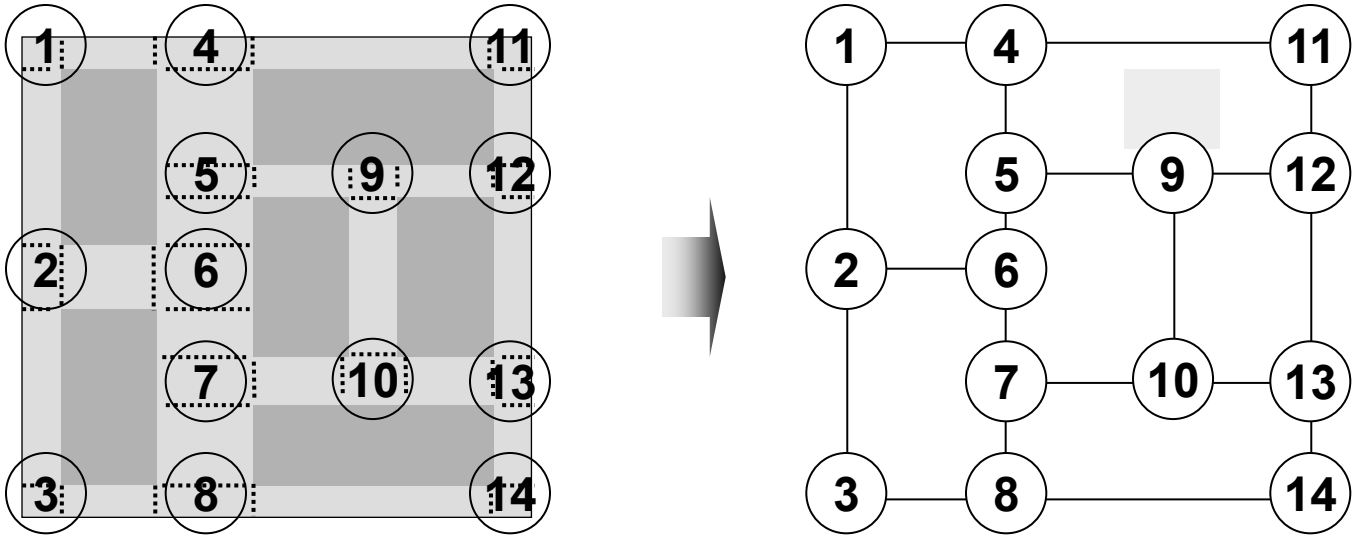
(a) real circuit with G-cells (b) grid graph for routing

Channel connectivity graph



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

Switchbox connectivity graph



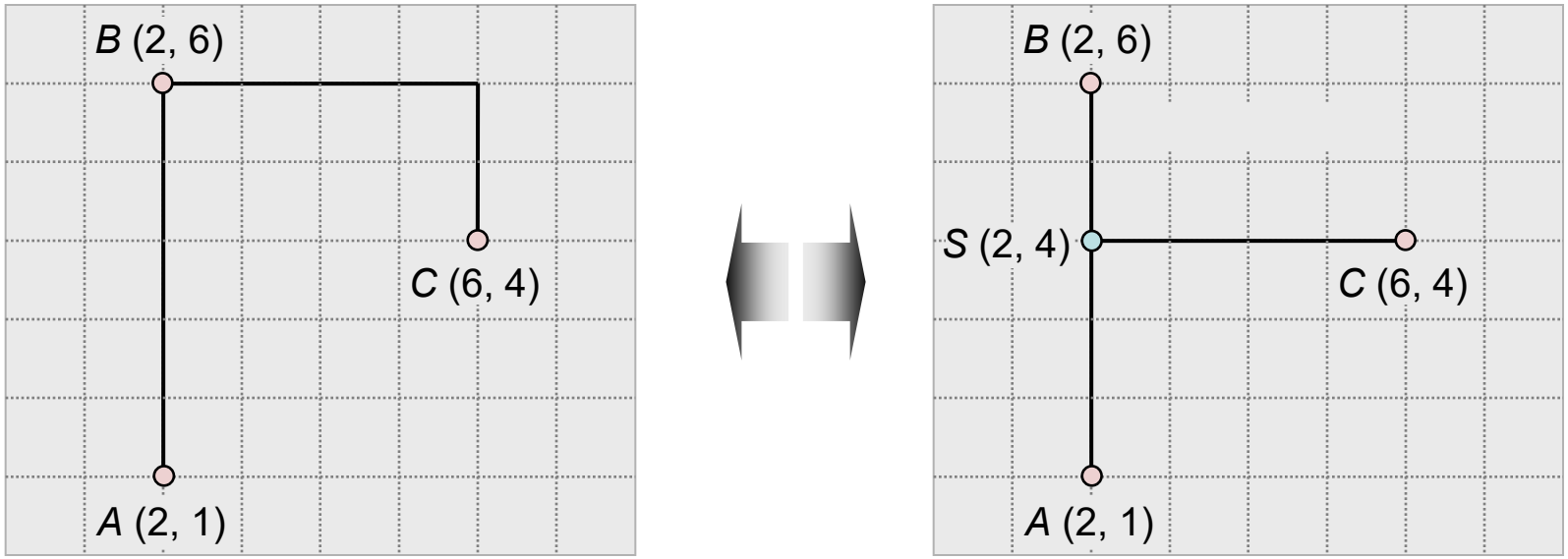
$G = (V, E)$, where the nodes $v \in V$ represent switchboxes
and an edge exists between two nodes if the corresponding switchboxes are on opposite sides of the same channel

5 min break

Global Routing Approaches

- Sequential approaches: one net at a time
  - Sensitive to ordering
  - Usually sequenced by
    - Criticality
    - Length (is it obvious whether longer first, or shorter first?)
    - Number of terminals
  - Two-terminal algorithms
    - Maze routing algorithms
    - Line probing
    - Graph shortest-path based algorithms
  - Multi-terminal algorithms
    - Steiner tree algorithms + decomposition into a sequence of subtree-to-subtree connections
- Concurrent approaches: simultaneously all nets
  - Computationally hard
  - Hierarchical methods used

Rectilinear Routing



Rectilinear minimum spanning tree (RMST)

Rectilinear Steiner minimum tree (RSMT)

- An RMST can be computed in $O(p^2)$ time, where p is the number of terminals in the net using methods such as Prim's Algorithm
- Prim's Algorithm builds an MST by starting with a single terminal and greedily adding least-cost edges to the partially-constructed tree

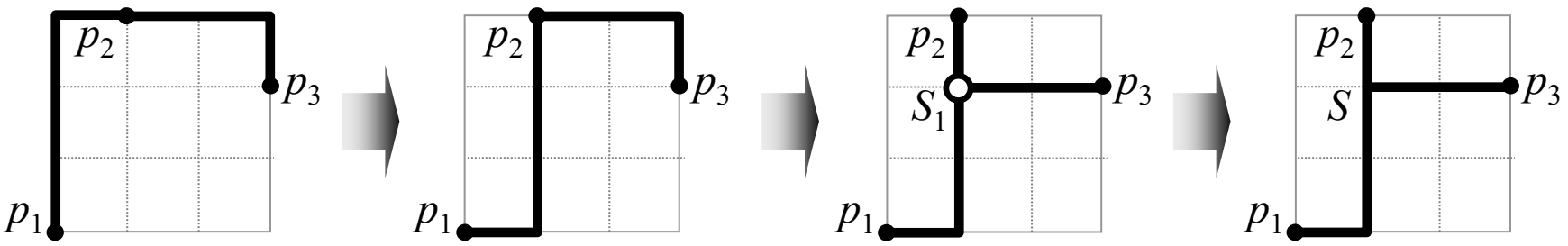
Rectilinear Routing

Characteristics of an RSMT

- An RSMT for a p-pin net has between 0 and $p - 2$ (inclusive) Steiner points
- The degree of any terminal pin is 1, 2, 3, or 4
  The degree of a Steiner point is either 3 or 4
- A RSMT is always enclosed in the minimum bounding box (MBB) of the net
- The total edge length L_{RSMT} of the RSMT is at least half the perimeter of the minimum bounding box of the net: $L_{RSMT} >= L_{MBB} / 2$
- Remember RSMT problem is NP Complete

Rectilinear Routing

Transforming an initial RMST into a low-cost RSMT



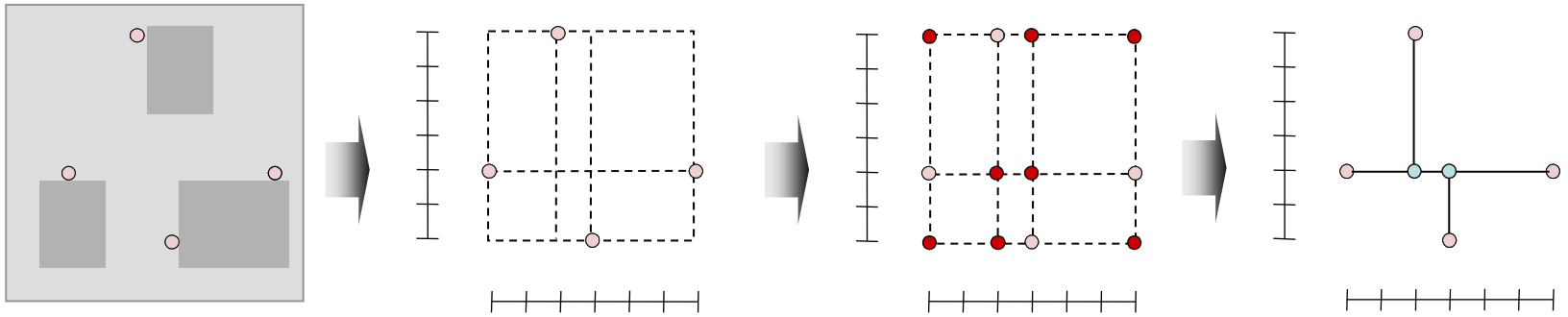
Construct *L*-shapes between points with (most) overlap of net segments

Final tree (RSMT)

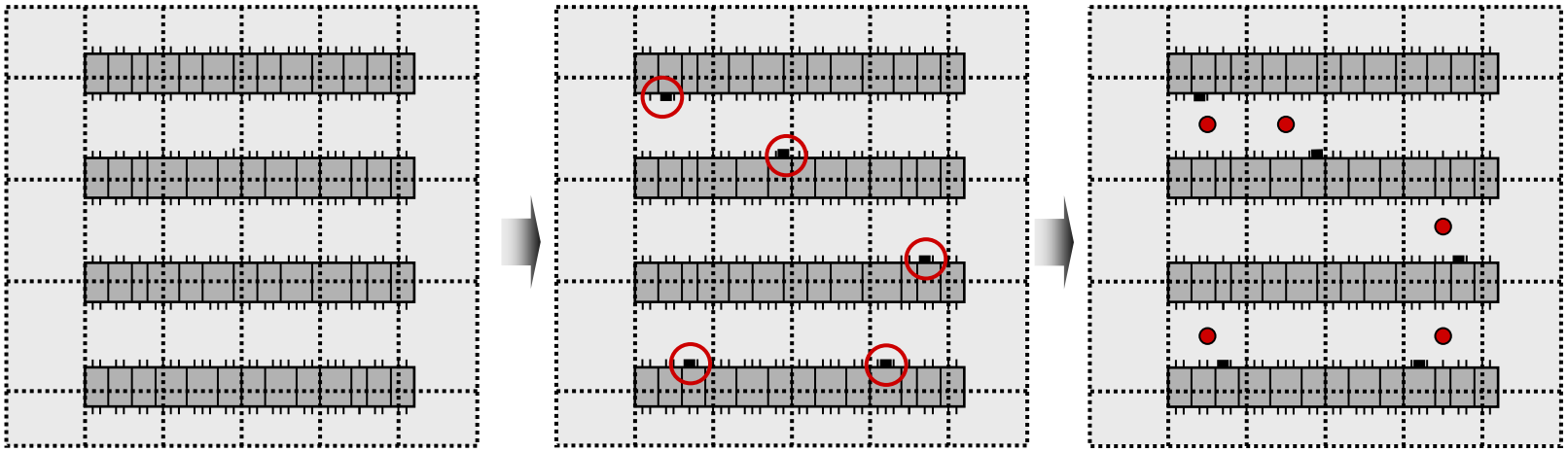
Hanan grid

- Adding Steiner points to an RMST can significantly reduce the wirelength
- Maurice Hanan proved that for finding Steiner points, it suffices to consider only points located at the intersections of vertical and horizontal lines
  that pass through terminal pins
- The Hanan grid consists of the lines $x = x_p, y = y_p$ that pass through the location (x_p,y_p) of each terminal pin p
- The Hanan grid contains at most (n^2-n) candidate Steiner points (n = number of pins), thereby greatly reducing the solution space for finding an RSMT

Hanan grid

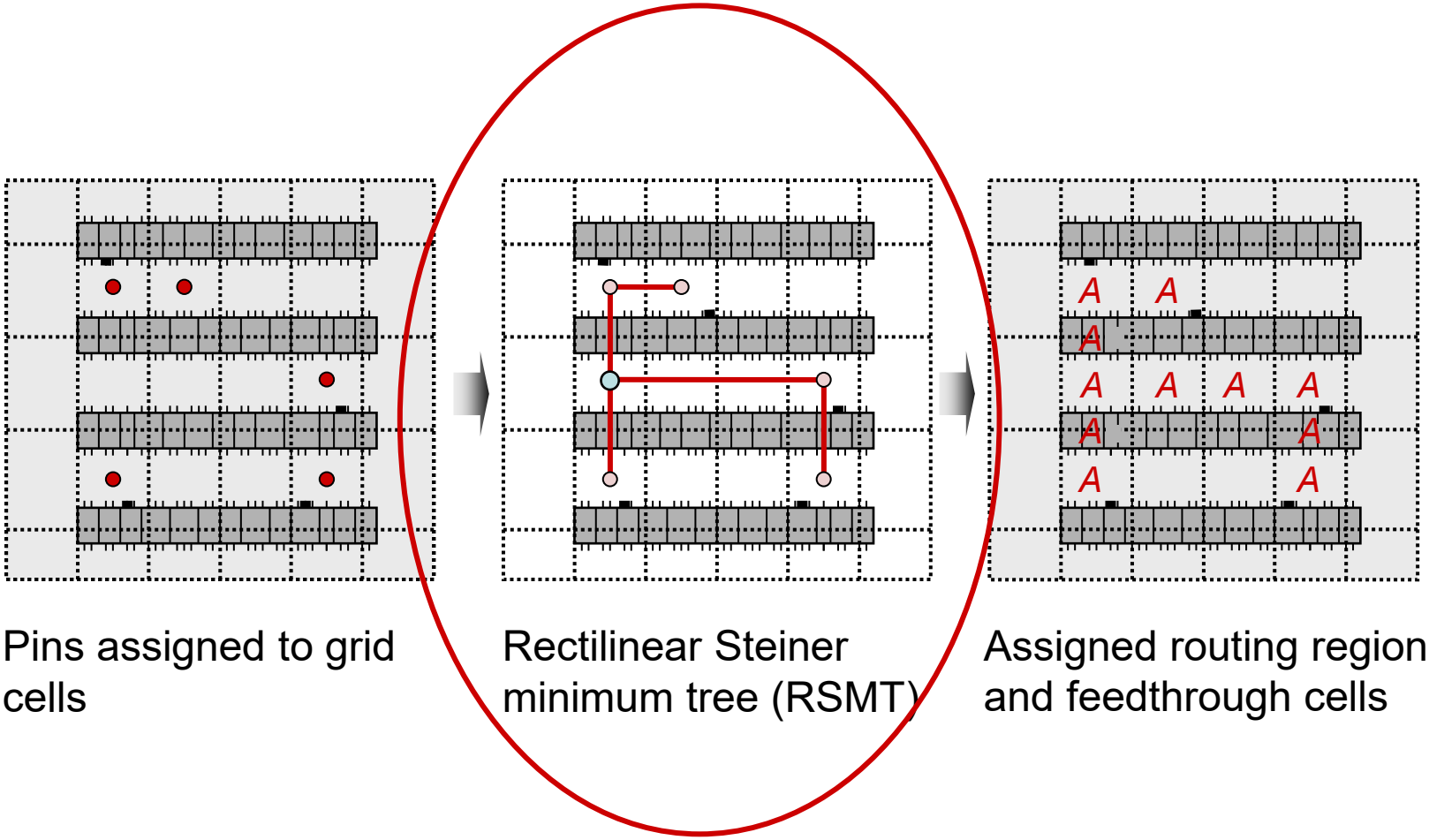


Example GR Flow



Definining routing regions Pin connections Pins assigned to grid cells

Example GR Flow



Pins assigned to grid cells

Rectilinear Steiner minimum tree (RSMT)

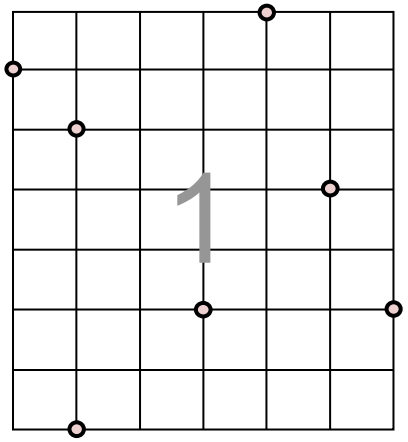
Assigned routing regions and feedthrough cells

A Sequential Steiner Tree Heuristic

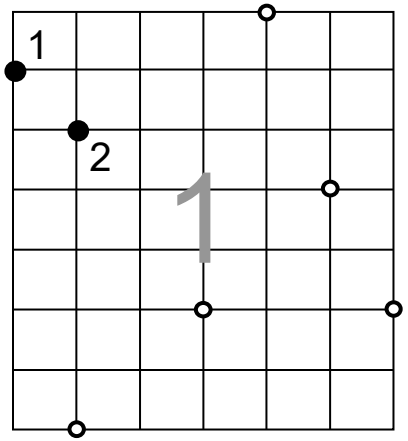
1. Find the closest (in terms of rectilinear distance) pin pair, construct their minimum bounding box (MBB)
2. Find the closest point pair (p_{MBB},p_C) between any point p_{MBB} on the MBB and p_C from the set of pins to consider
3. Construct the MBB of p_{MBB} and p_C
4. Add the L-shape that p_{MBB} lies on to T (deleting the other L-shape). If p_{MBB} is a pin, then add any L-shape of the MBB to T.
5. Goto step 2 until the set of pins to consider is empty

Resembles Prim's algorithm for MST

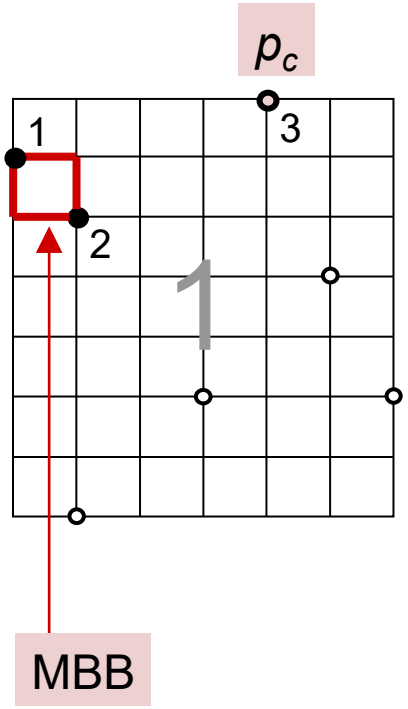
Example Sequential Steiner Tree Heuristic



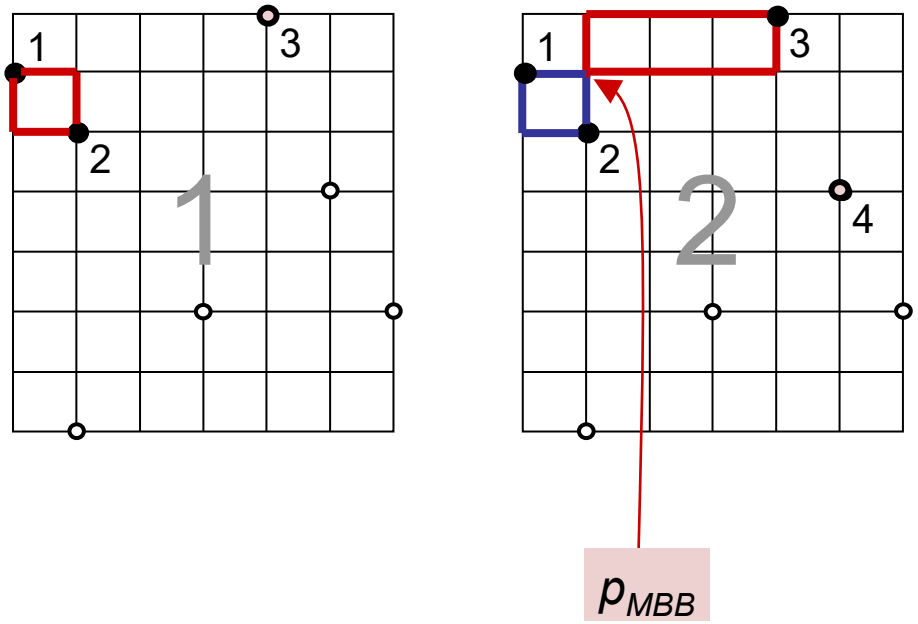
Example Sequential Steiner Tree Heuristic



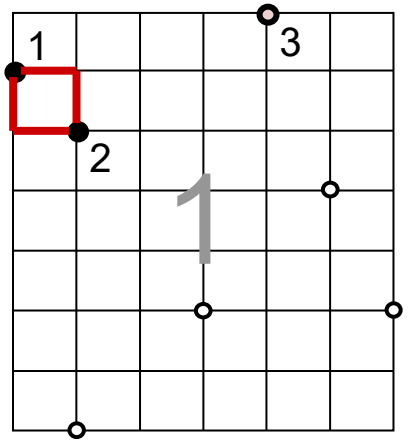
Example Sequential Steiner Tree Heuristic

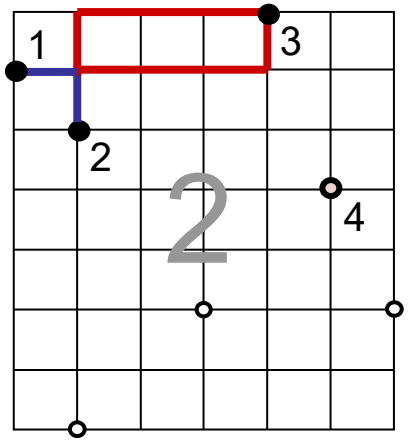


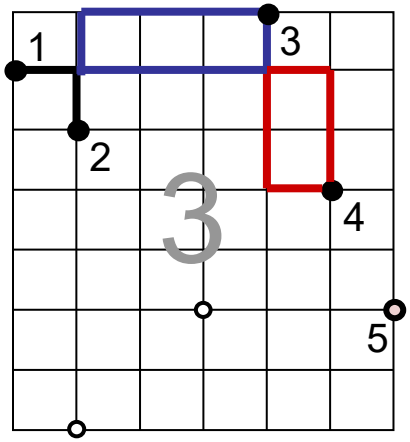
Example Sequential Steiner Tree Heuristic



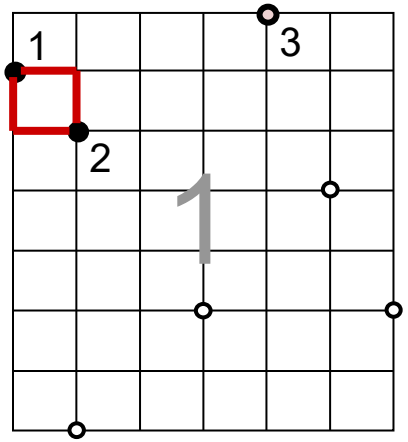
Example Sequential Steiner Tree Heuristic

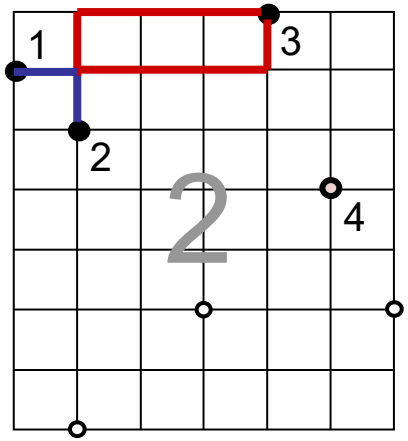


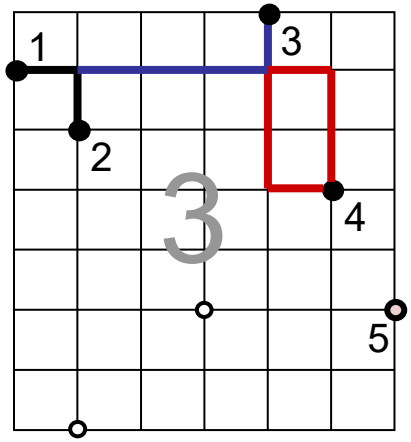


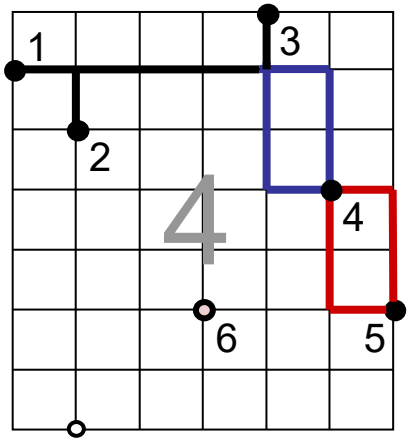


Example Sequential Steiner Tree Heuristic

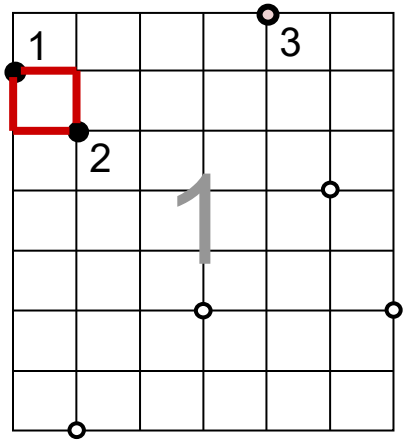


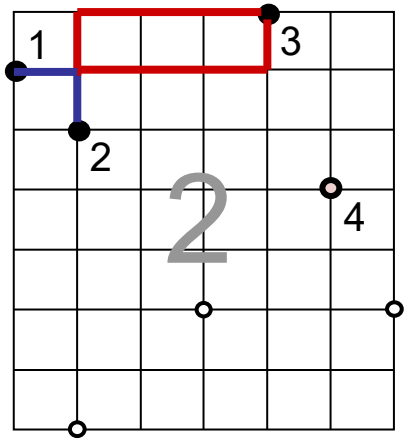


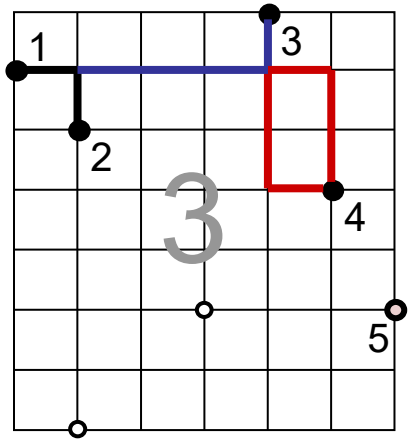


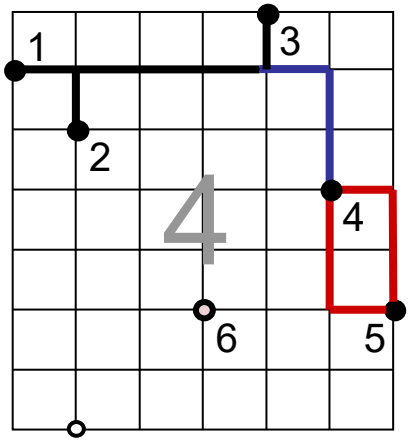


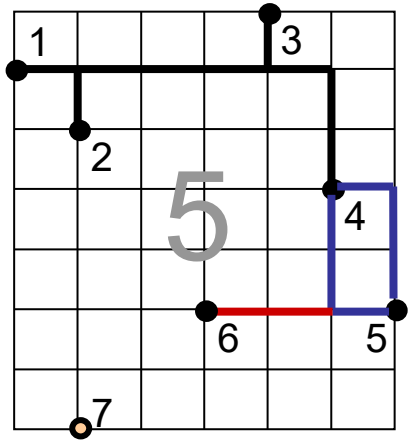
Example Sequential Steiner Tree Heuristic



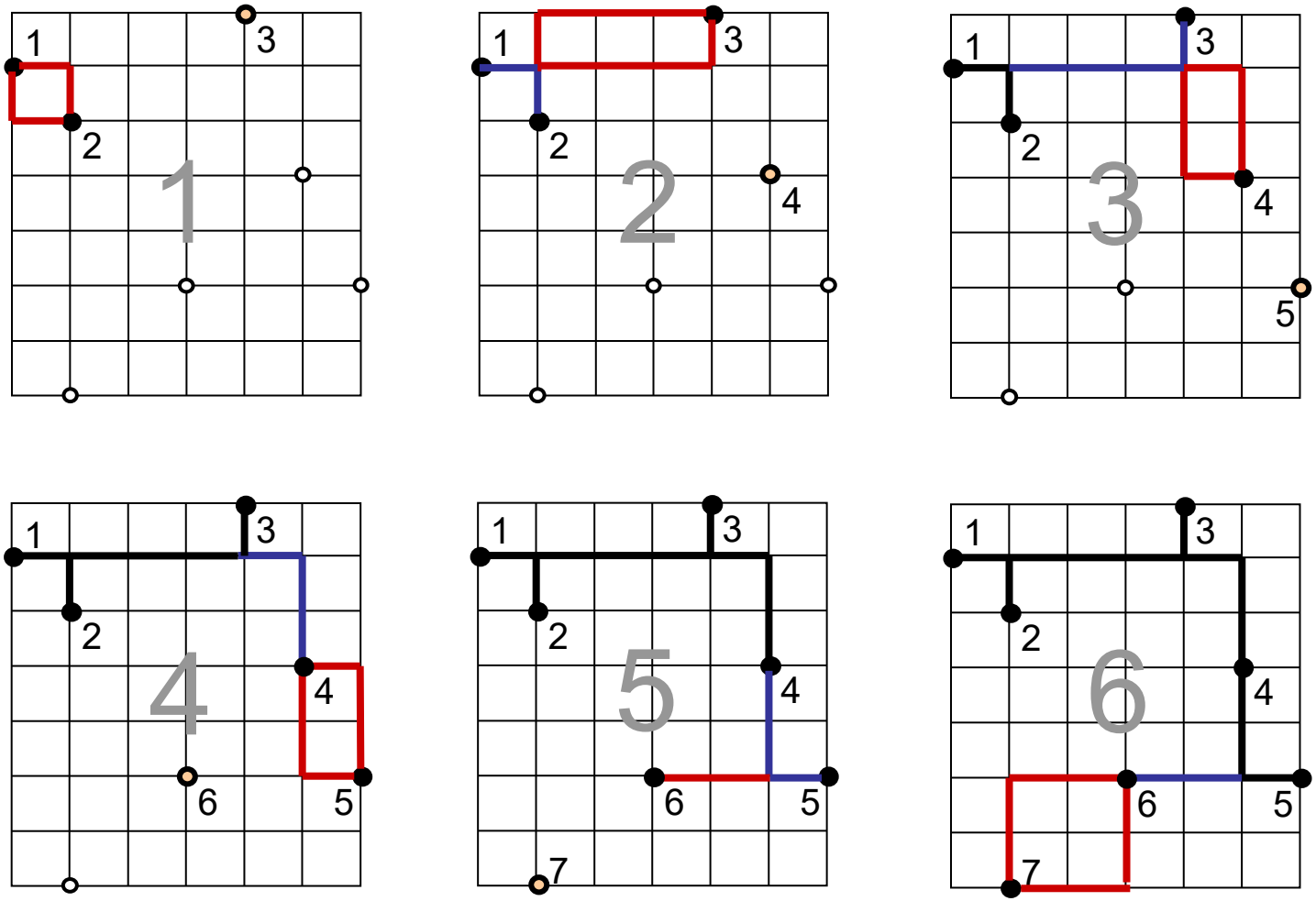




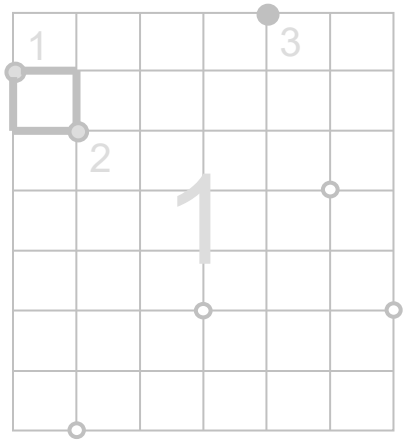


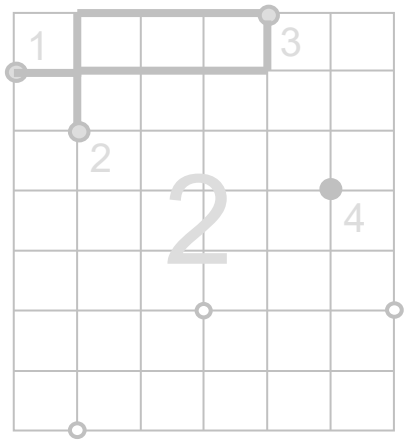


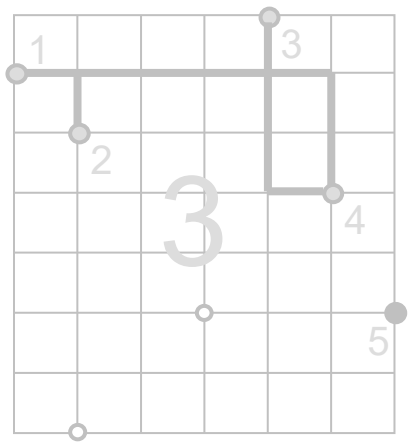
Example Sequential Steiner Tree Heuristic

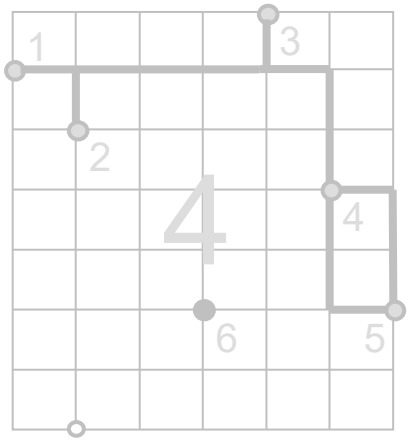


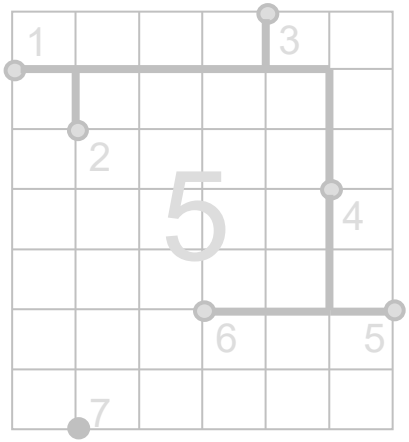
Example Sequential Steiner Tree Heuristic

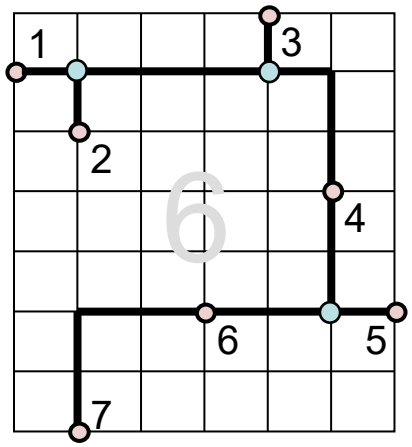






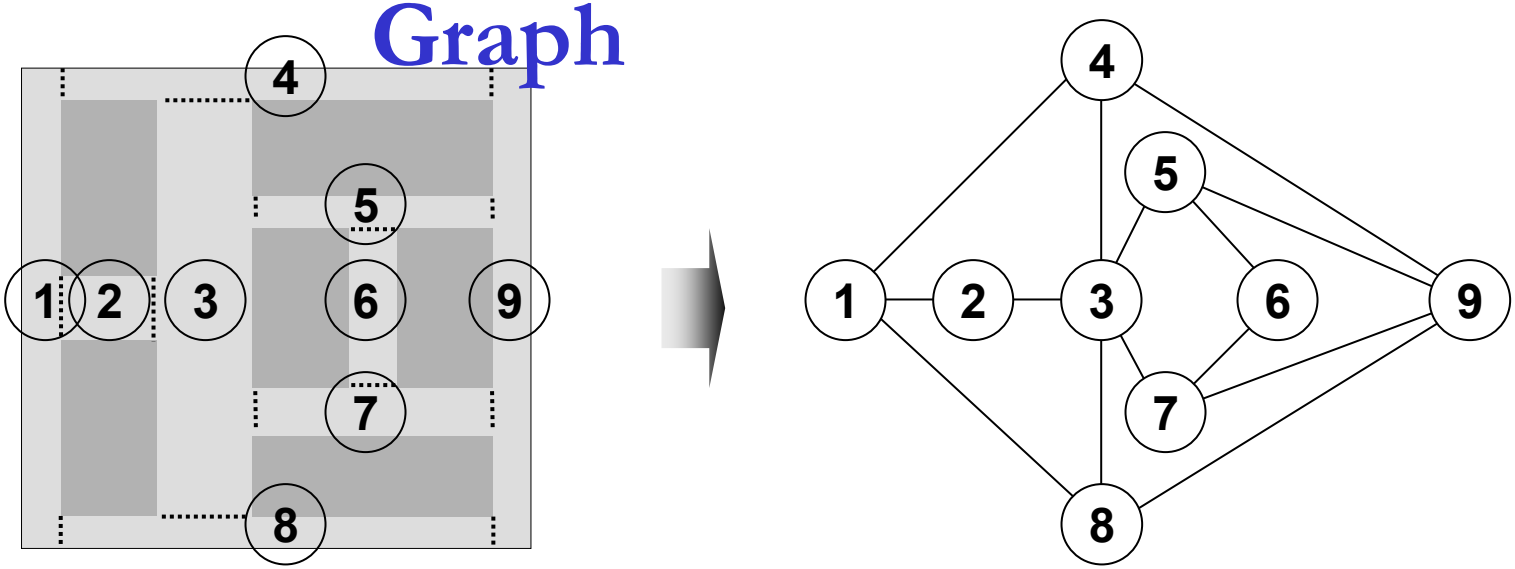




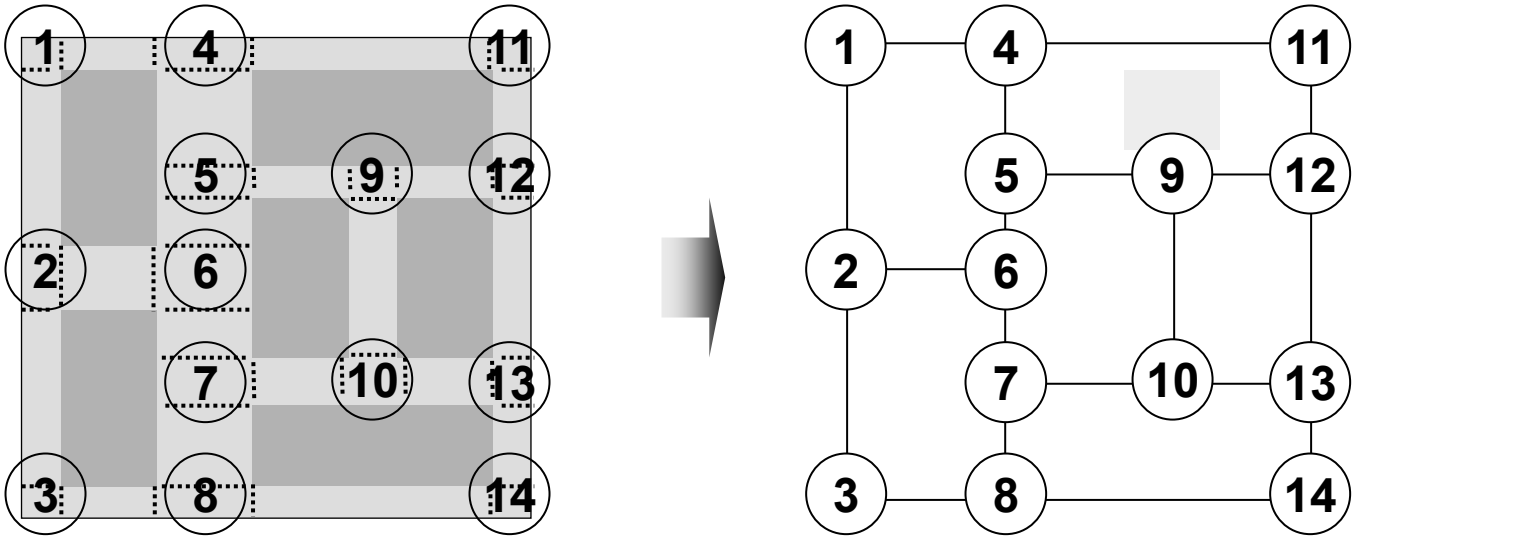


Global Routing in a Connectivity Graph

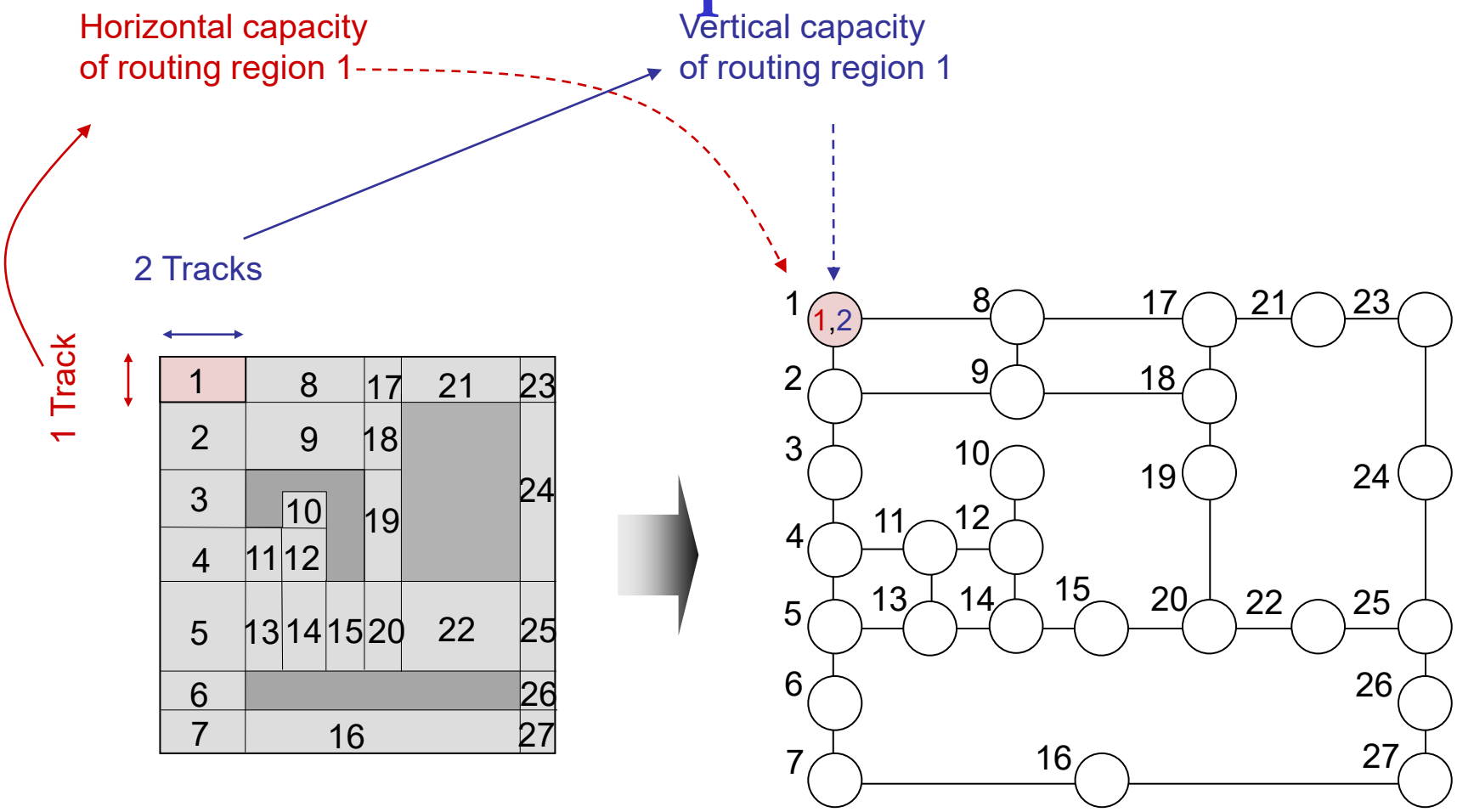
Channel connectivity graph



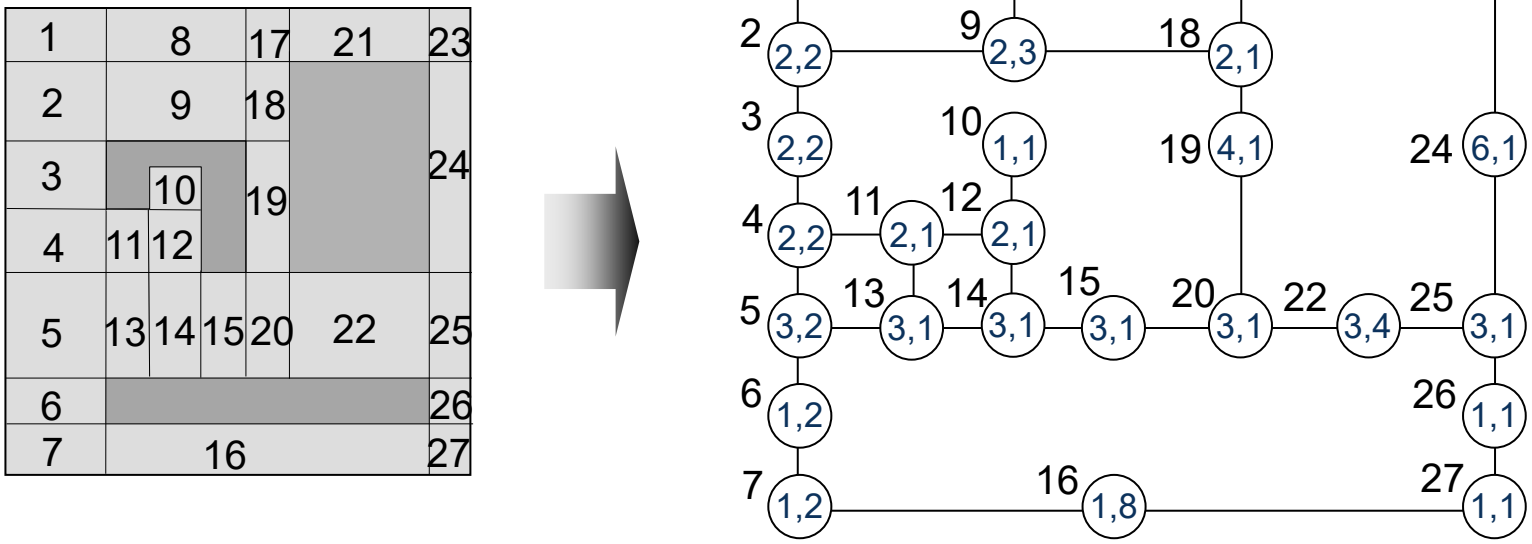
Switchbox connectivity graph



Global Routing in a Connectivity Graph

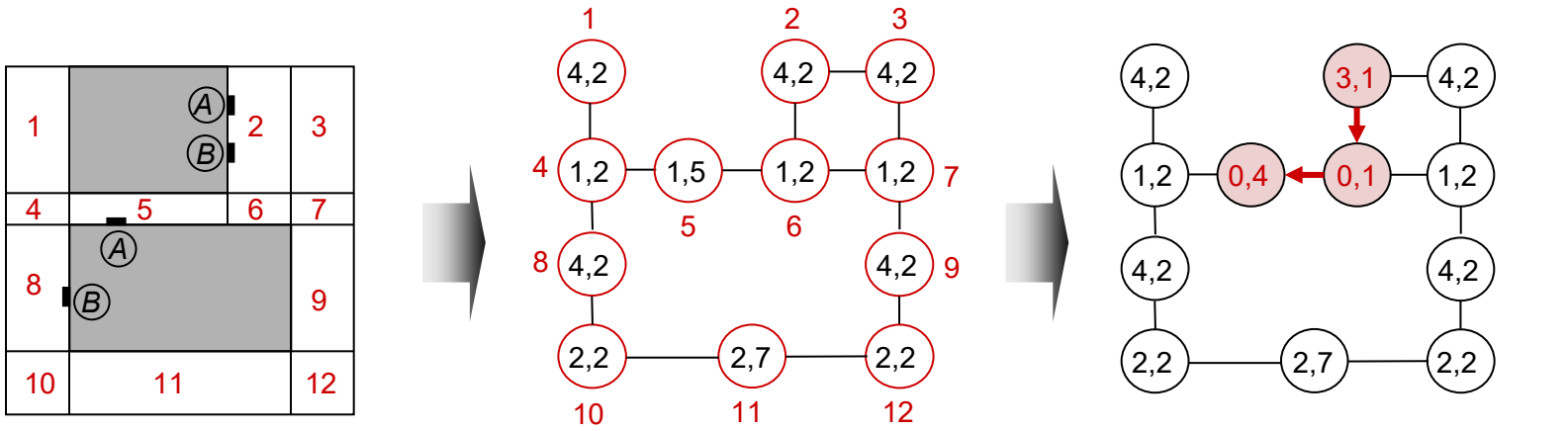


Global Routing in a Connectivity Graph



Global Routing in a Connectivity Graph

- Combines switchboxes and channels, handles non-rectangular block shapes
- Suitable for full-custom design and multi-chip modules



Routing regions

Graph representation

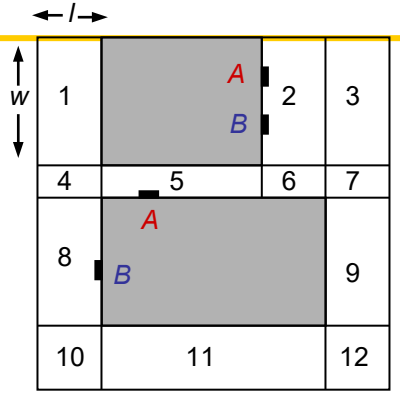
Graph-based path search

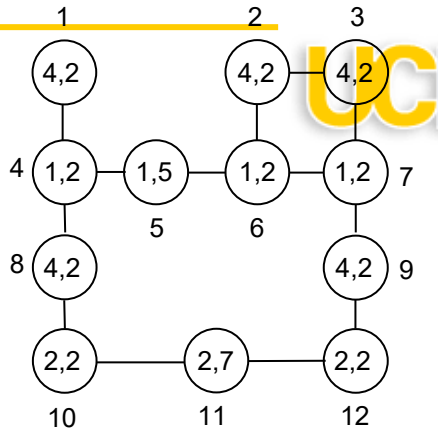
Basic Algorithm Overview

1. Define routing regions
2. Define connectivity graph
3. Determine net ordering
4. Consider each net
   a) Free corresponding tracks for net's pins
   b) Decompose net into two-pin subnets
   c) Find shortest path for subnet connectivity graph
   d) Update routing capacities
5. If there are unrouted nets, goto Step 4, otherwise END

Example

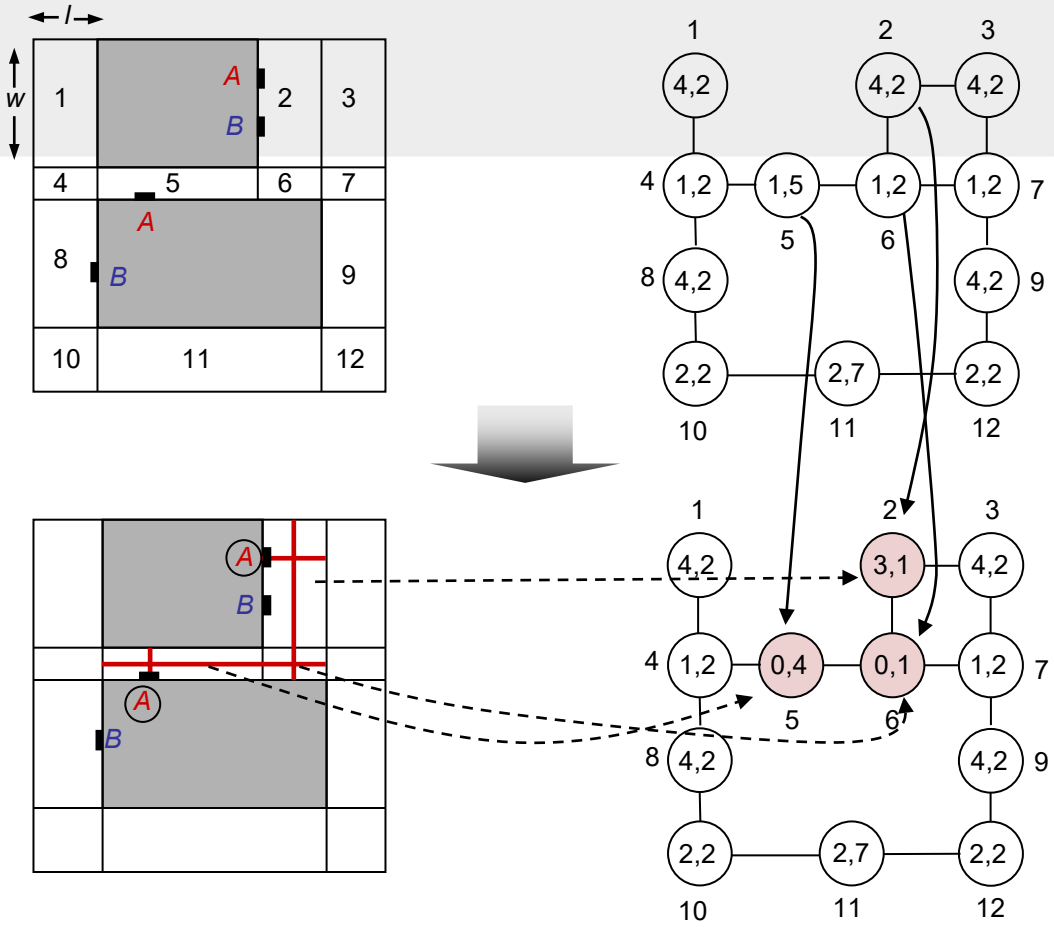
Global routing
of the nets A-A and B-B





Example

Global routing
of the nets A-A and B-B



Example

Global routing
of the nets A-A and B-B