CAD for VLSI

References

- Algorithm for VLSI Design Automation
  - Sabih H Gerez
- High - Level Synthesis Introduction to Chip & System Design
  - Daniel D Gajski, Nikil D Dutt, Allien C-H Wu, Steve Y-L Lin
- Logic synthesis & Verification Algorithms
  - Gary D Hachtel, Fabio Somenzi
- Graph Theory with application to Engg & Computer Science
  - Narsingh Deo

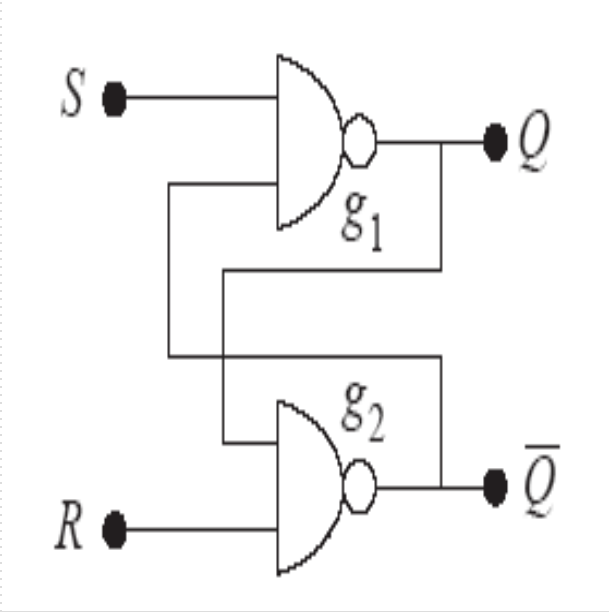
Placement (Ch. 7 Gerez)

Contents

- Circuit Representation
- Graph Model
- Standard cell placement
- Building block placement
- Constructive & Iterative Placement
- Partitioning Techniques
- Constructive & Iterative Partitioning

Circuit Representation

- Cell
- Port
- Net



SR - Latch

Circuit Representation

Cell

- Basic building block of circuit eg. NAND gate
- Name of the cell
- List of I/ps & o/ps
- Electrical properties
- Switching delays
- Layout properties, width & height
- Cell library can be pre-designed cells or the standard cells or the library of cells designed by the designer.

Circuit Representation

Port

- A point at which a connection between a wire and a cell is established
- Cell structure needs to have access to its port.
- Set of I/p ports
- Set of O/p ports
- Parent cell
- Connected net

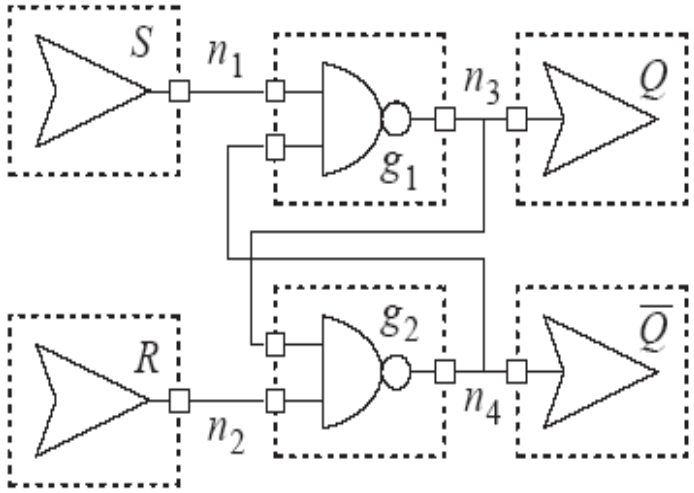
Circuit Representation

Net

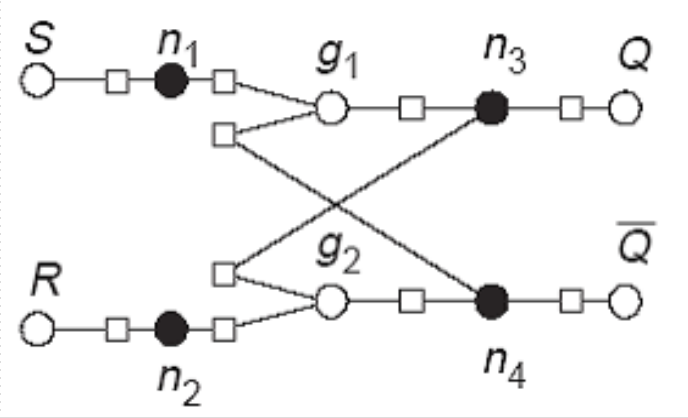
- The wire that electrically connects two or more ports is a net.

Graph Model

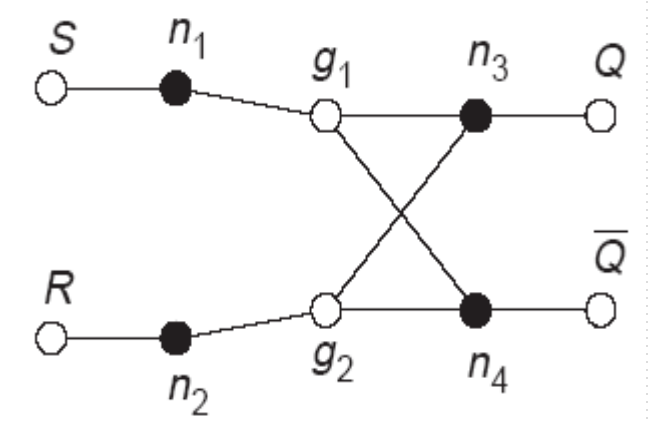
- 3 distinct set of vertices
  - Cell set
  - Port set
  - Net set
- 2 edge sets
  - Connecting cells with the port
  - Connecting nets with the port
- Note edges never connect vertices of same type & neither do they connect nets with cells.



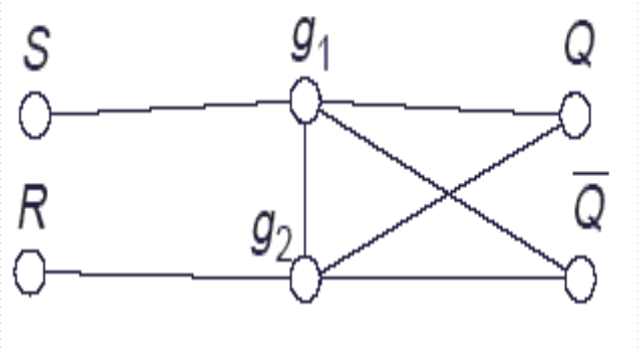
Graph Model



Tripartite graph model



Bipartite graph model



Clique model

Placement

- Placement is the problem of automatically assigning correct positions for predesigned cells on a chip/silicon area, such that some cost function is optimized.

- Different ways of designing create different placement problems. The most important of these are:

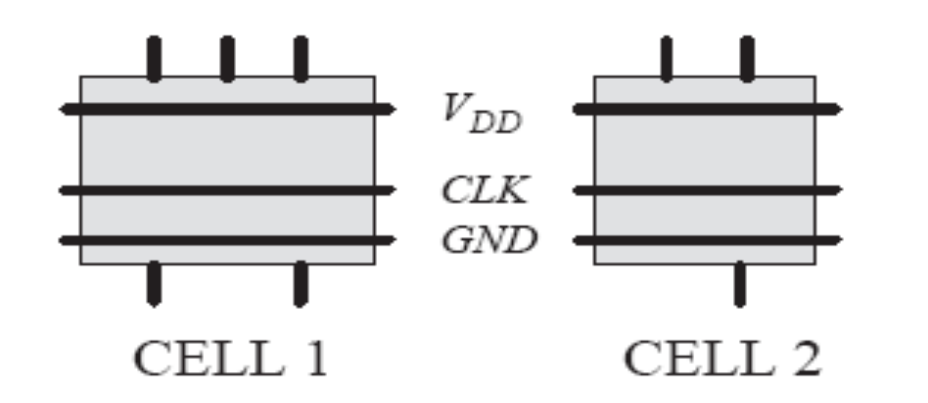
  - Standard-cell placement;

  - Building-block placement;

  - A combination of the above.

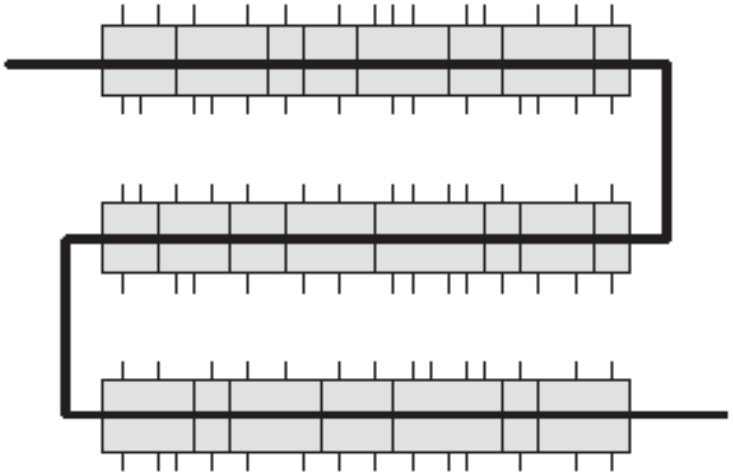
Standard - Cell Placement

Standard cells are designed in such a way that power and clock connections run horizontally through the cell at fixed positions and other I/O leaves the cell from the top or bottom sides.

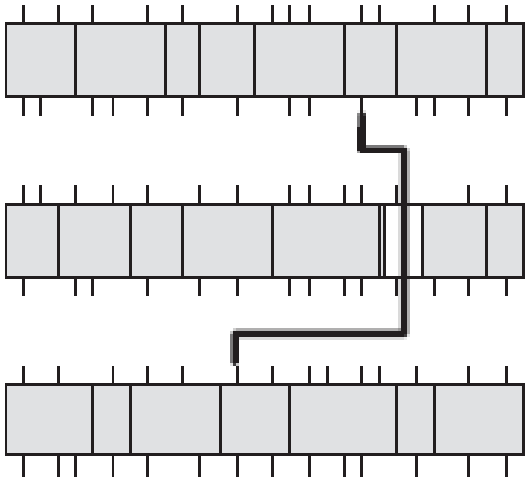


Standard - Cell Placement

The cells are placed in rows

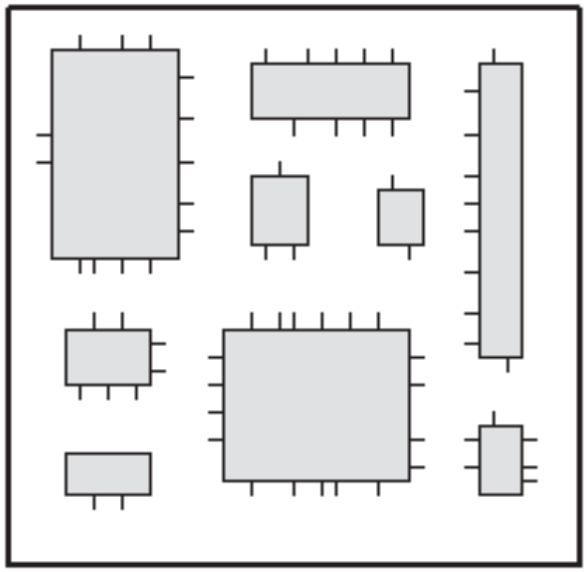


Sometimes feed through cells are added to ease wiring.



Building Block Placement

The cells to be placed have arbitrary shapes.



Cost Function

What does a placement algorithm try to optimize?

- The total area,
- The total wire length,
- The number of horizontal/vertical wire segments crossing a line.

Constraint:

- The placement should be routable (no cell overlaps; obeying wiring capacitances).
- In some designs: timing constraints (some wires should always be shorter than a given length).

Relation with Routing

- Ideally, placement and routing should be performed simultaneously as they depend on each other's results. This is, however, too complicated.

- In practice placement is done prior to routing. The placement algorithm estimates the wire length of a net using some metric.

PLACEMENT ALGORITHMS

- The placement problem is an NP-complete problem, heuristics are used in placement algorithms.
- Placement algorithms are classified into two categories:

Constructive algorithms

- Once the position of a cell is fixed, it is not modified anymore.

Iterative algorithms

- Intermediate placements are modified in an attempt to improve the cost function.

PLACEMENT ALGORITHMS

- Most approaches combine both elements:
- Constructive algorithms are used to obtain an initial placement.
- The initial placement is followed by an iterative improvement phase

Routing

Contents

- Local routing
- Characterization
- Area Routing
- Channel Routing
- Vertical Constraint Graph
- Horizontal Constraint Graph
- Global Routing
- Standard cell
- Building Block

Introduction to Routing

- Interconnection of cells
- Specification
  - Position of terminals
  - Netlist - indicates which terminals need to be interconnected.
  - Area - Available for routing in each layer

Introduction to Local Routing

- Routing is performed in two stages
- Global or Loose Routing
  - Decides which wiring channel a wire will run.
- Local Routing
  - Fixes precise path a wire will take.

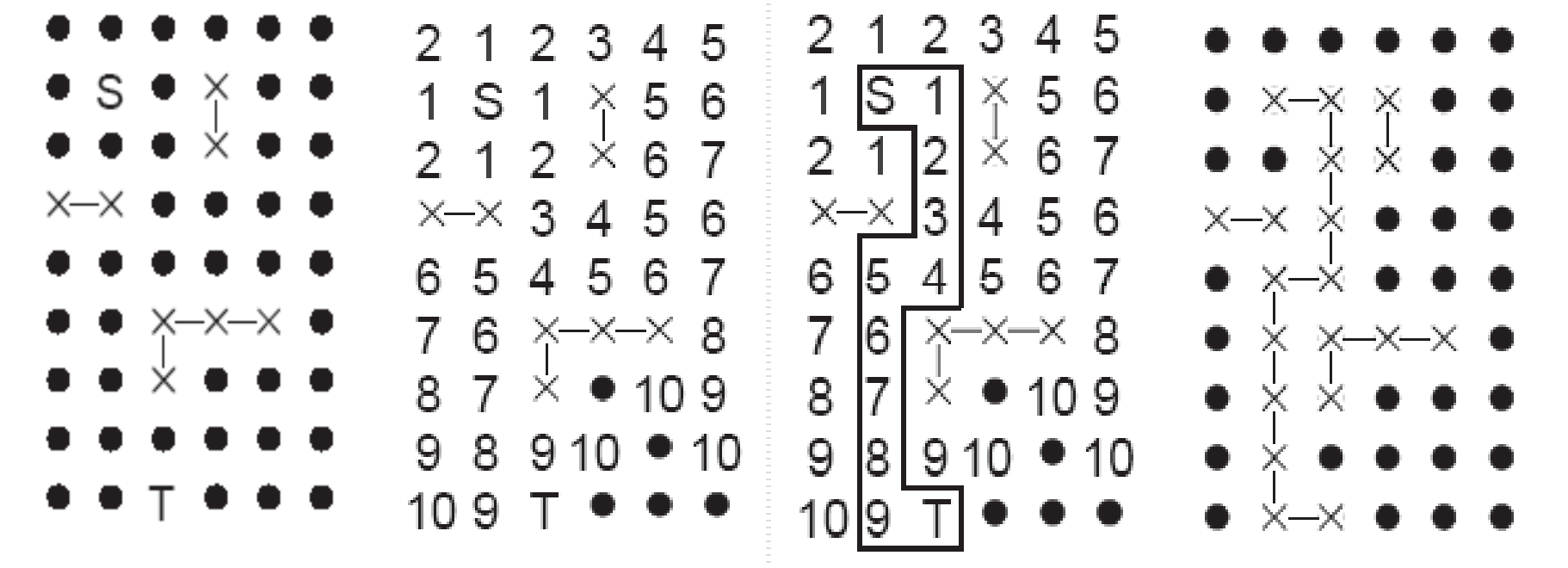
Types of Local Routing Problems – Parameters

- Number of wiring layers
- Orientation of wiring layers
- Gridded or gridless routing
- The Presence or Absence of Obstacles
- The position of the terminals
- Terminal with fixed or floating position
- Permutability of terminals
- Electrically Equivalent terminals

Area Routing - Lee's Algorithm

- The Lee algorithm is a classical routing technique, it is the basis of many routing programs. Also called as 'path connection' or 'maze routing' algorithm
- The one-layer version is presented.
- The routing area is a grid; terminals can be located anywhere in the area available for routing → area routing.
- The algorithm looks for connections in the presence of obstacles.

Area Routing - Lee's Algorithm



A two-point connection is realized by propagating a wave front from the **source** terminal outwards until the **target** terminal is reached and using backtracking for finding the shortest connection

3 Steps : Wave propagation, Backtracking, Cleanup

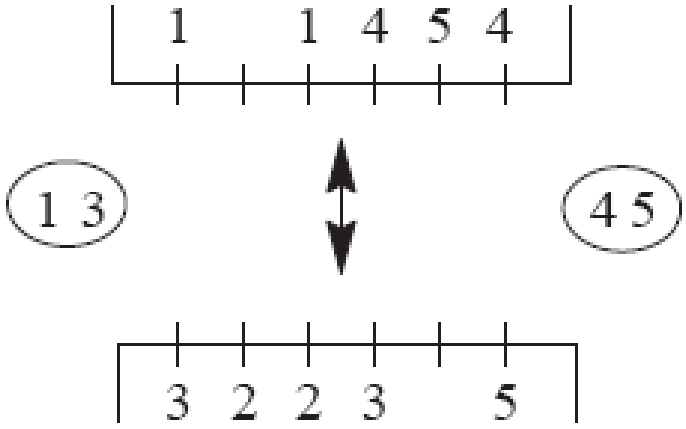
Area Routing - Lee's Algorithm

- Multi-terminal nets: first two terminals are connected, this connection is the target for the wave propagation from the third terminal, etc.
- A routed net is an obstacle for the next nets
- The algorithm always finds a connection if it exists
- For two-terminal nets, this connection is the shortest possible path; for multi-terminal nets optimality is not guaranteed
- It can be generalized for multiple layers: wave front expansion need to be in three dimensions
- Its time complexity and space complexity is $O(n^2)$
- The quality of its result strongly depends on the ordering of the nets

Channel Routing

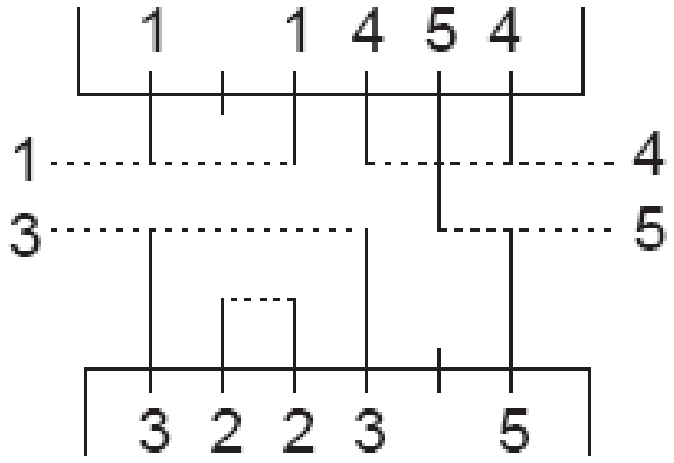
Channel routing is characterized by:

- A Rectangular routing area
- The Top and Bottom rectangle boundaries contain terminals with fixed positions
- The Left and Right boundaries of the channel have terminals with floating positions
- The primary goal is to minimize the height of the routing area. Secondary goal is the minimization of total wire length and the number of via's

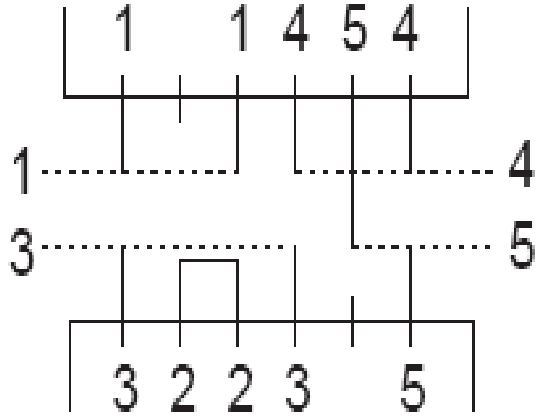


Channel Routing - Example

Solution in a reserved-layer *model*

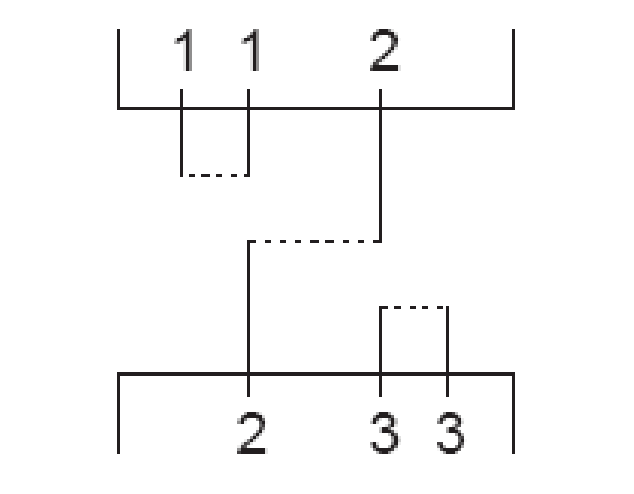


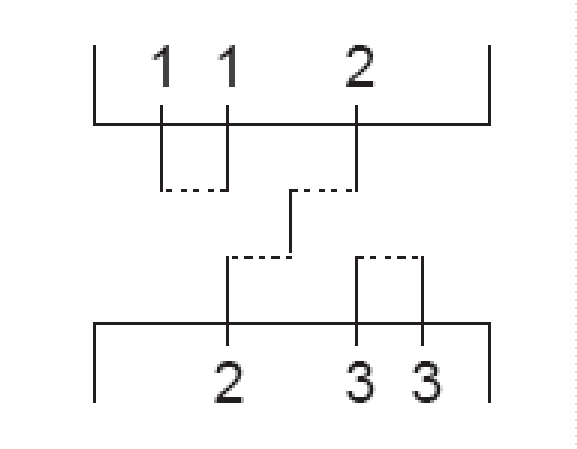
Solution in a nonreserved-layer *model*



Channel Routing – Use of Doglegs

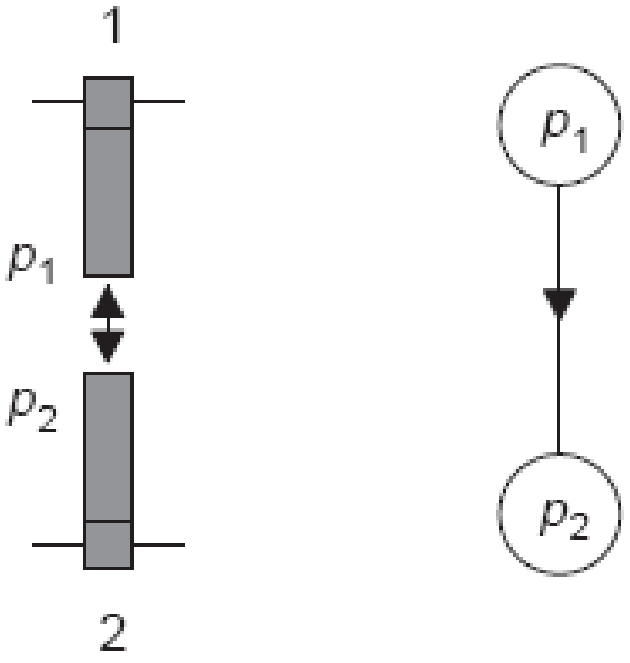
- Some routing models assume that each net only uses a single horizontal segment.
- The use of multiple horizontal segments, doglegs, per net can lead to better results (but larger search space).





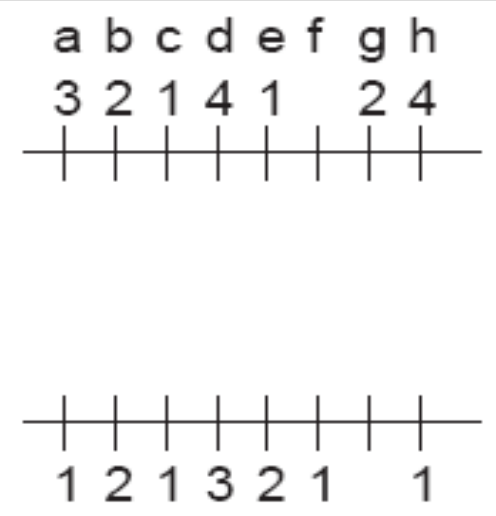
Vertical Constraints

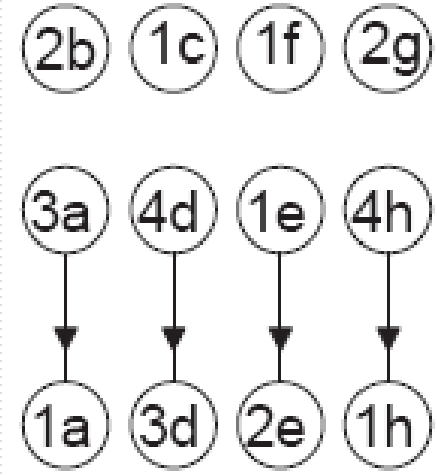
- Two terminals located above each other give rise to a vertical constraint
- the vertical segment connected to the top terminal cannot overlap with the vertical segment connected to the bottom terminal.

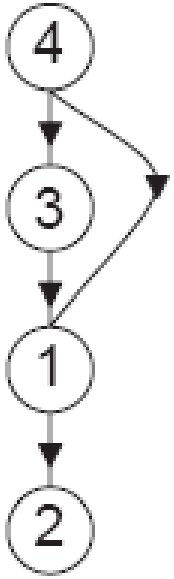


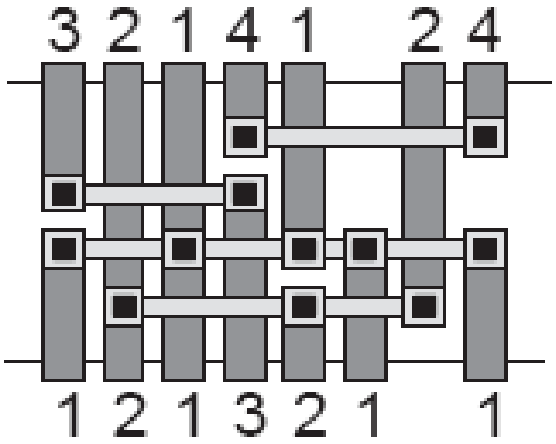
Vertical Constraints

Vertical constraints can be combined into a vertical constraint graph under the assumption that each net will use one horizontal segment.



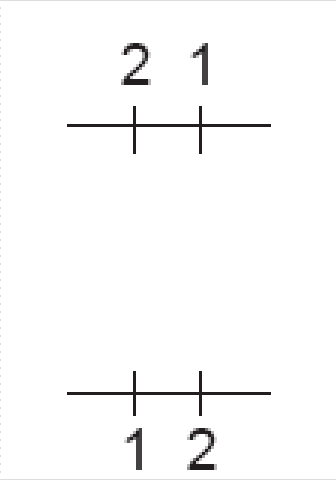


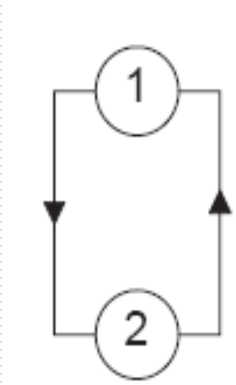


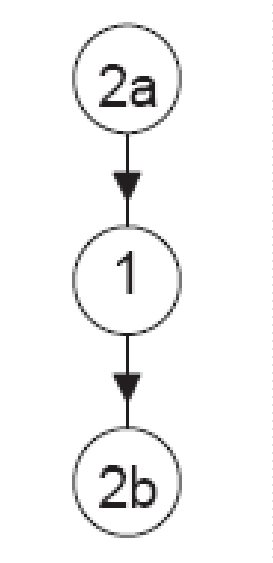


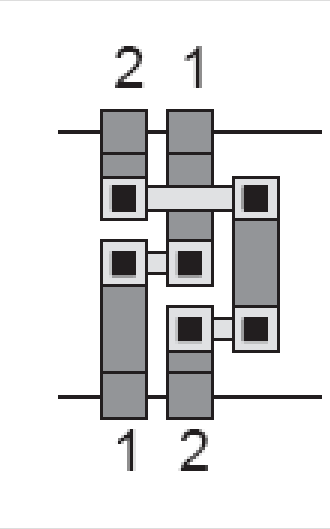
Vertical Constraints

Cyclic constraints must be resolved by splitting horizontal segments.









Horizontal Constraint

- The horizontal segments of different nets cannot be located on the same track. This is called a horizontal constraint.

- The combination of horizontal and vertical constraints makes the channel routing problem difficult.

Channel routing models - classical

- All wires run along orthogonal grid with uniform separation
- There are two wiring layers
- Horizontal segments are put on one layer and the vertical ones on the other.
- For each net, a single horizontal segment is used, with the vertical segments connecting it to all the terminals.

Left Edge Algorithm

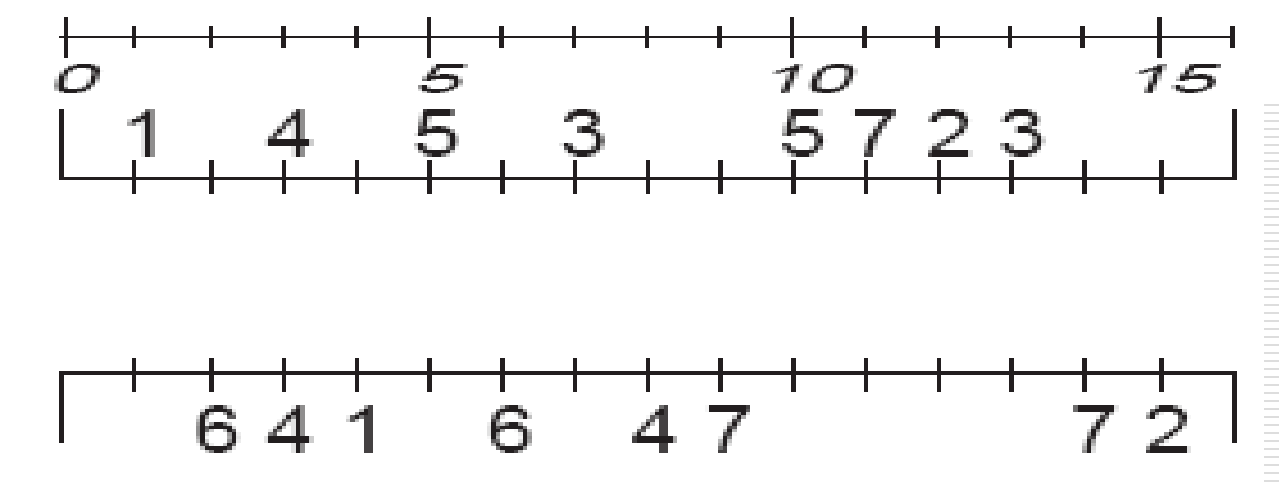
Channel Density

- Local density $d(x)$: the number of nets that cross the channel at column position x.
- Channel density $d\max$: $d\max = \max d(x)$.
- The channel density is a lower bound on the number of rows (one horizontal segment per net).
- Channel routing problems without vertical constraints can be solved efficiently by the left-edge algorithm.

Left Edge Algorithm

- Problem: given a set of segments (intervals) $[x_{i\min}, x_{i\max}]$, put non-overlapping segments on the same track such that the number of rows is minimal.
- Solution: Sort the intervals by their first coordinate and fill the rows by the first fitting interval.
- Remark: The simple left-edge algorithm solves the problem optimally (in polynomial time).

Left Edge Algorithm…



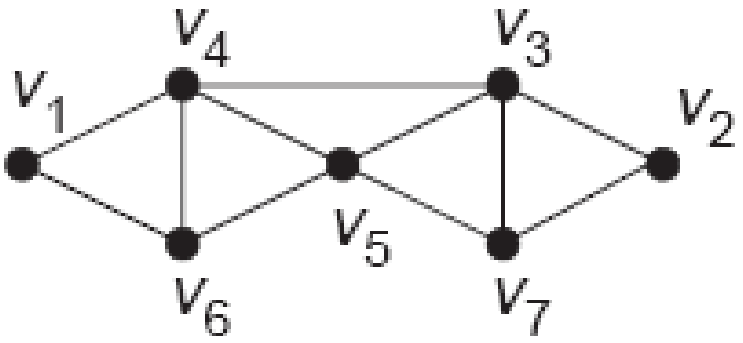
Intervals

$i_1 = [1, 4]$ $i_2 = [12, 15]$ $i_3 = [7, 13]$ $i_4 = [3, 8]$

$i_5 = [5, 10]$ $i_6 = [2, 6]$ $i_7 = [9, 14]$

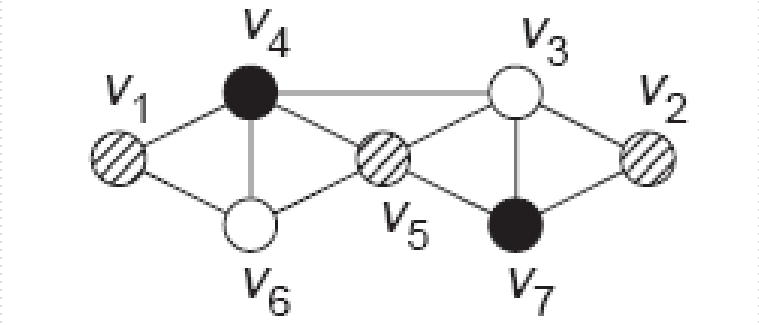
Left Edge Algorithm…

- There is a vertex for each interval.
- Vertices corresponding to overlapping intervals are connected by an edge.



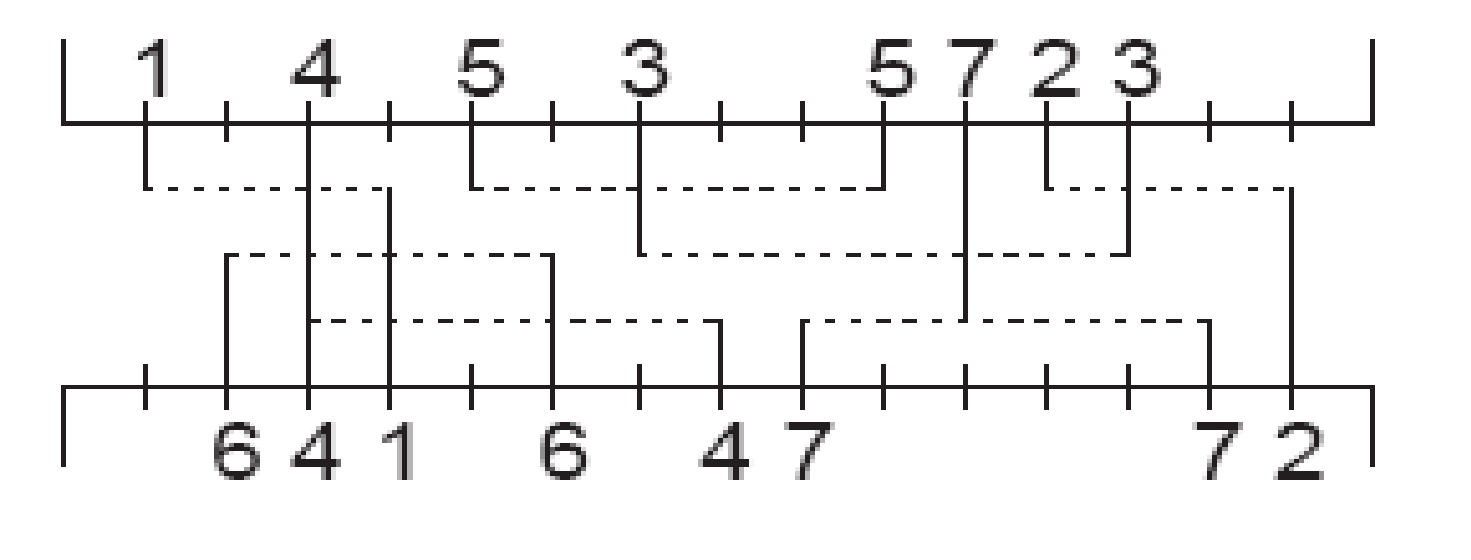
Consider the intervals:

$i_1 = [1, 4]$, $i_4 = [3, 8]$, $i_6 = [2, 6]$,
$i_3 = [7, 13]$, $i_5 = [5, 10]$, $i_7 = [9, 14]$
$i_2 = [12, 15]$.



- Solving the track assignment problem is equivalent to finding a minimal vertex coloring of the graph.

Left Edge Algorithm…

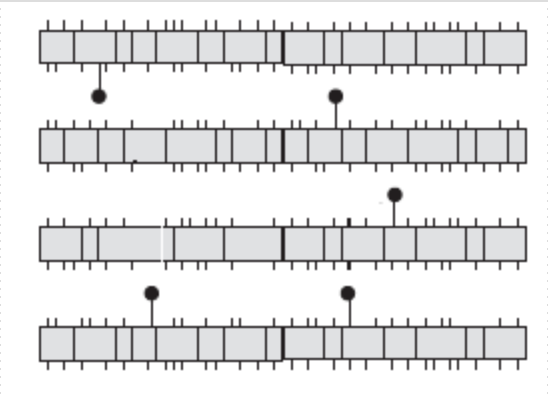


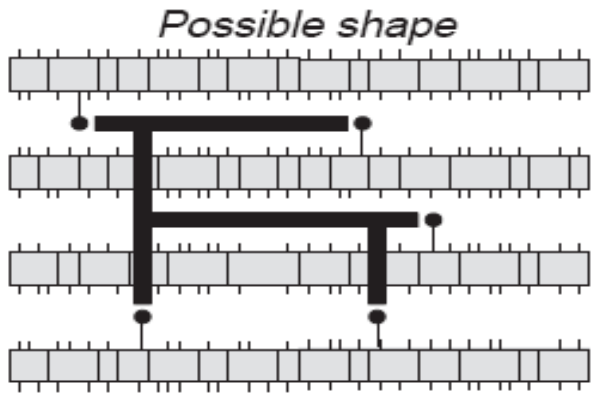
$i1 = [1, 4]$, $i5 = [5, 10]$, $i2 = [12, 15]$,
$i6 = [2, 6]$, $i3 = [7, 13]$,
$i4 = [3, 8]$, $i7 = [9, 14]$.

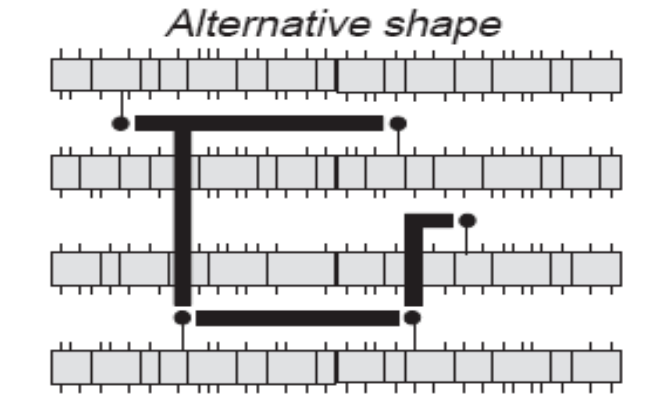
Global Routing

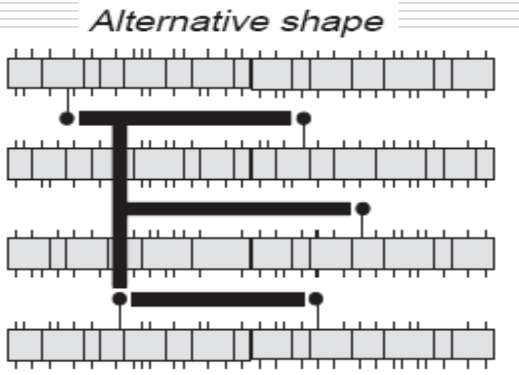
- Decides the distribution of interconnections across available channels.
- Then all the connections are established by solving the local routing problems for each channel separately
- This is the process of roughly fixing the shapes of the connections for each net.
- The channel density is used to estimate the global routing quality.
- **Standard cell**
  - Dedicated feedthrough cells,
  - Built-in feedthrough wires,
  - Unused metal layer

Standard cell









Time Driven Routing

- With the decrease of feature sizes in CMOS, timing is becoming increasingly important in VLSI design. Timing models for global routing are:

- **Lumped model:**
  - The entire wiring shape is considered a single resistor and capacitor; timing optimization amounts to reducing the total length of the wire.

- **Transmission-line model:**
  - The distinct wiring shapes are separately modeled, making the signal arrive at different times at the terminals
  - Timing optimization amounts to minimizing the wire length from the source terminal to the critical sink.

Building block

- Problem more complex than in standard-cell layout.
- Identification of channels not always obvious: the channel definition problem.
- Channel ordering is important to ensure routability.

Layout Compaction

Contents

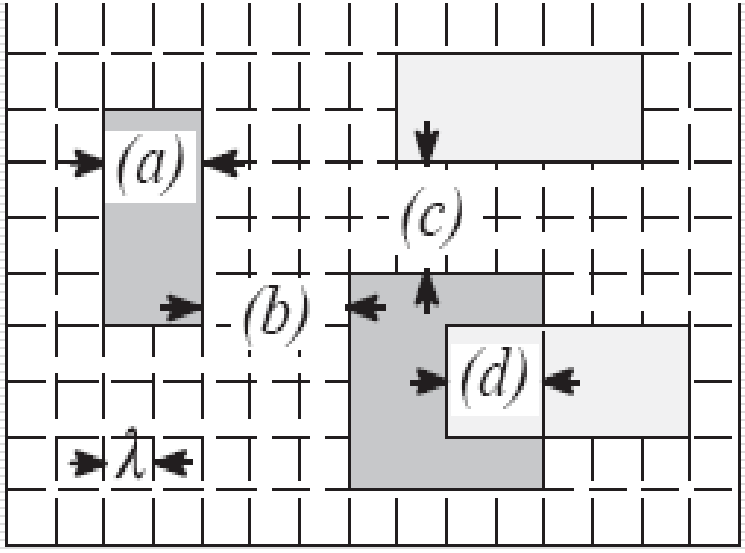
- Design Rules
- Symbolic Layout
- Min distance inequality
- Max distance inequality
- Longest Path Algorithm
- Liao - Wong Algorithm
- Bellman - Ford Algorithm
- Some Issues in Compaction

Layout Compaction

- At the level of mask patterns used for the fabrication of the circuit, a final optimization can be applied to remove redundant space.
- This optimization is called layout compaction.

Design Rules

- Restrictions on the mask patterns to increase the probability of successful fabrication.

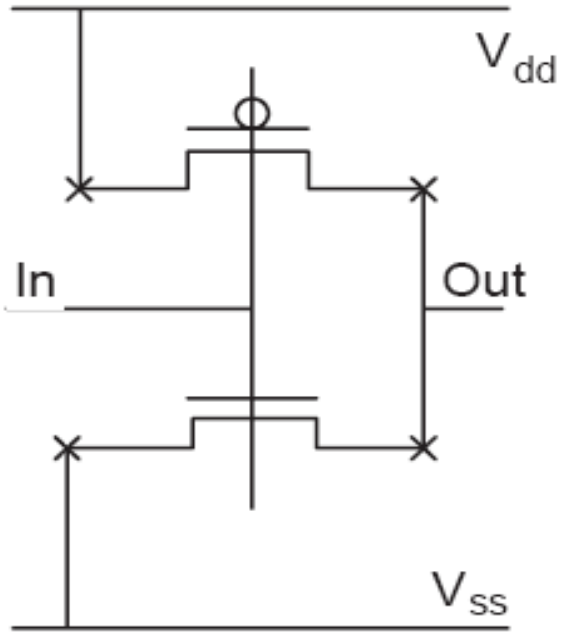


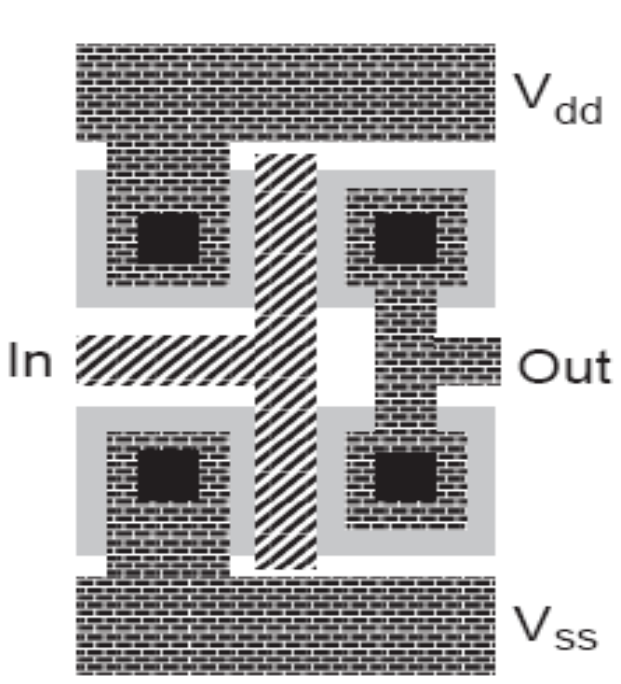
- Patterns and design rules are expressed in Lambda. The most common design rules are
  - Min-width rules :valid for a mask pattern of a specific layer;(a).
  - Min-separation rule : between mask patterns of the same layer or different layers;(b), (c).
  - Min-overlap rule :mask patterns in different layers; (d).

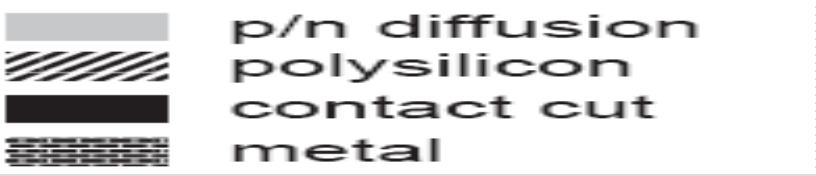
Symbolic Layout

- A layout is symbolic when symbols are used to represent the circuit elements and their interconnections, leaving out the exact geometric details of layers.
  - Symbolic layout fixes the topology of a layout, i.e., relative positions and shapes of different layers.
  - The length, width or layer of a wire or other layout element might be left unspecified.
  - Mask layers not directly related to the functionality of the circuit do not need to be specified, e.g. n-well, p-well.

Symbolic Layout example







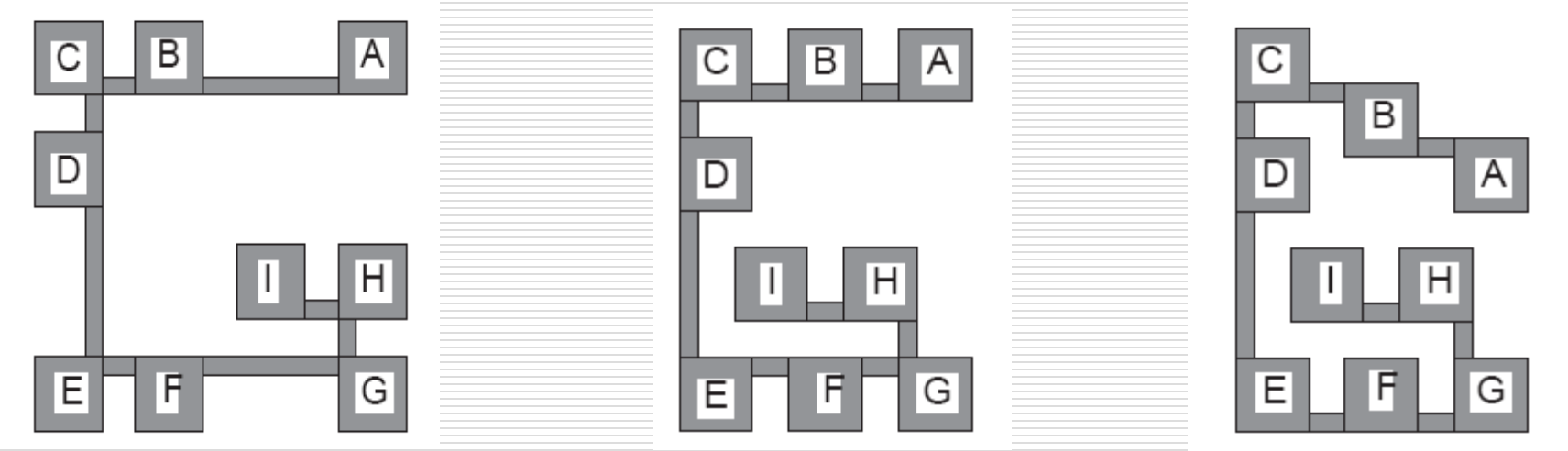
Applications of compaction…

- A compaction program or compactor can generate layout at the mask level. It attempts to make the layout as dense as possible.
- Applications of compaction:
  - Area minimization: removing redundant space in layout at the mask level.
  - Layout compilation: generation of geometric/mask-level layout from symbolic layout.
  - Fixing design rules: automatic removal of design-rule violations.
  - Technology migration: converting mask-level layout from one technology to another.

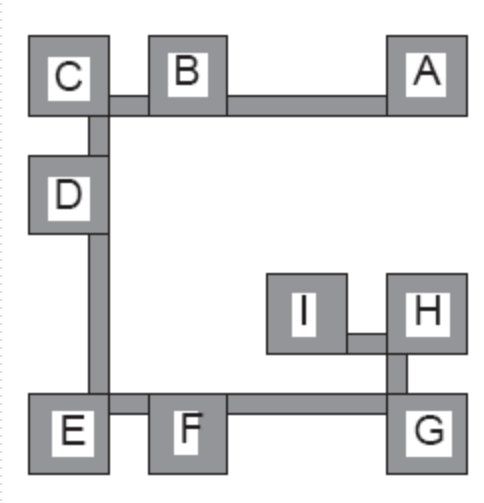
Layout Compaction – Redundant space removal

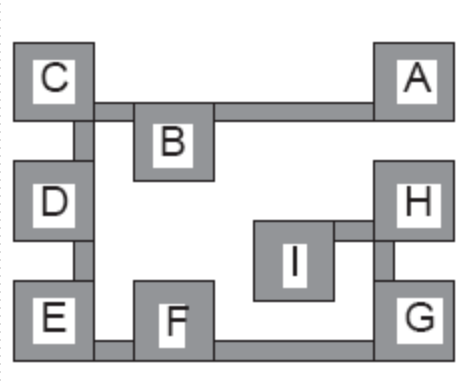
- Dimension:
  - 1-dimensional (1D): layout elements only move or shrink in one dimension (x or y). Often sequentially performed first in the x-dimension and then in the y-dimension (or vice versa).
  - 2-dimensional (2D): layout elements move and shrink simultaneously in two dimensions.
- Complexity:
  - 1D-compaction can be done efficiently; 2D-compaction is NP-hard.

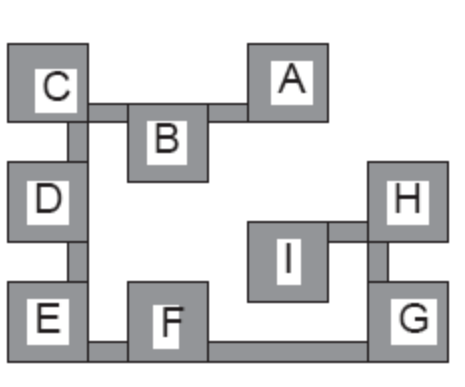
1D COMPACTION: X followed by Y



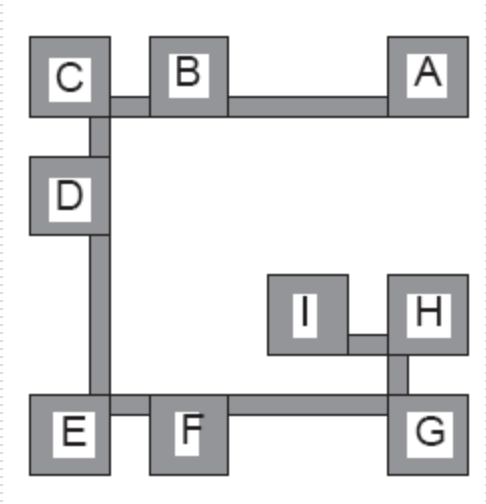
1D COMPACTION: Y followed by X

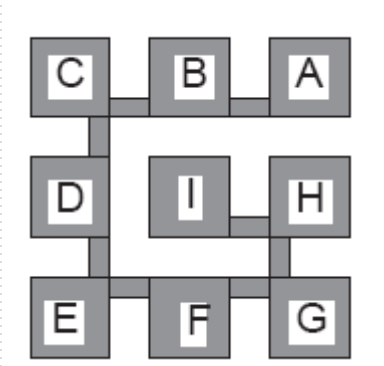






2D COMPACTION



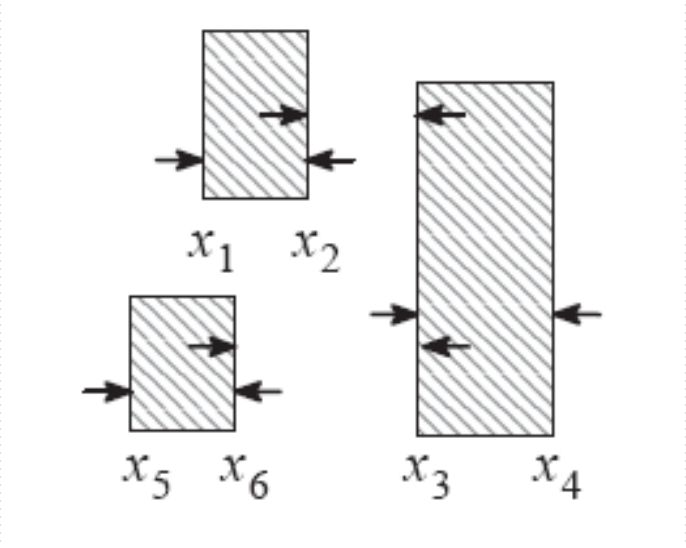


Graph Theoretical Formulation

- Minimum-distance design rules can be expressed as inequalities.

For example:

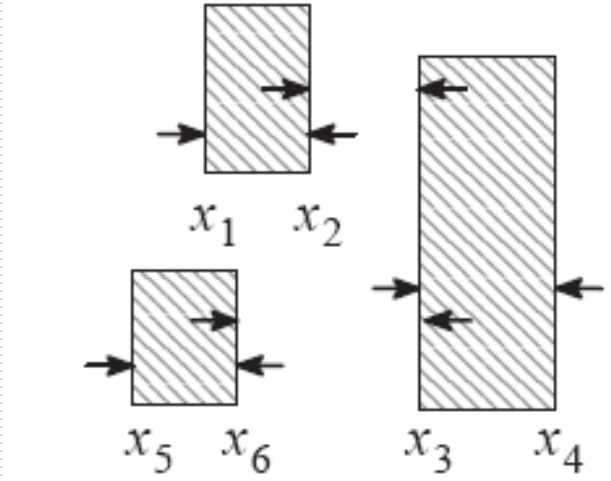
$$x_2 - x_1 \geq a$$
$$x_3 - x_2 \geq b$$
$$x_3 - x_6 \geq b$$

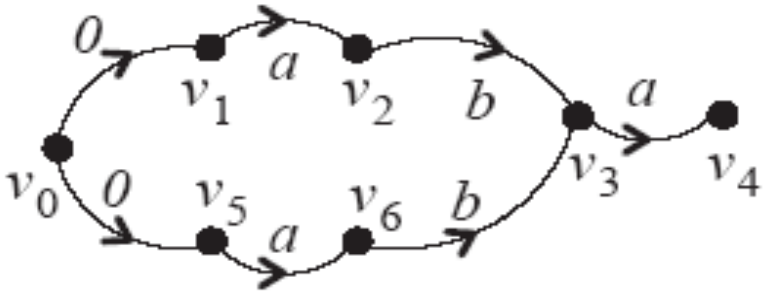


Min width → a
Min separation → b

Graph Theoretical Formulation

- The inequalities can be used to construct a constraint graph G(V, E):
- There is a vertex v_i for each variable x_i.
- For each inequality $x_j - x_i \geq d_{ij}$ there is an edge (v_i, v_j) with weight d_{ij} .
- There is an extra source vertex, $v0$; it is located at x = 0; all other vertices are at its right.
- If all the inequalities express minimum-distance constraints, the graph is acyclic. It is a DAG, a directed acyclic graph.





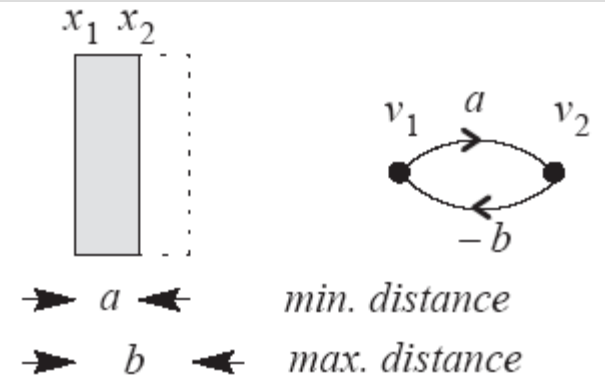
Maximum - Distance Constraint

- Sometimes the distance of layout elements is bounded by a maximum, e.g. when the user wants a maximum wire width.
- A maximum distance constraint gives an inequality of the form:

$$x_j - x_i \leq c_{ij}$$

$$x_i - x_j \geq -c_{ij}$$

- Consequence for the constraint graph: backward edge (v_j, v_i) with weight $d_{ji} = -c_{ij}$
- The graph is not acyclic anymore.



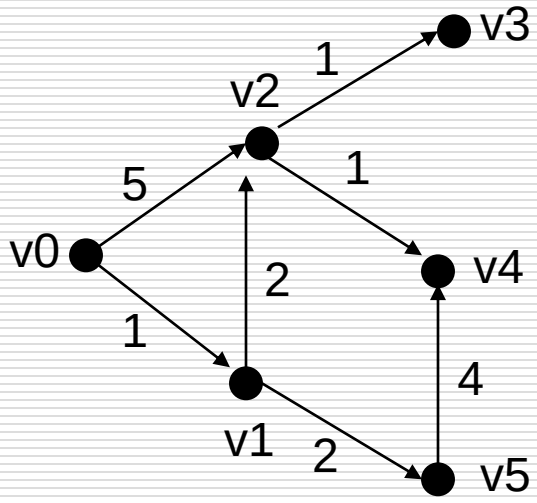
Longest path algorithm

- The length of the longest path from the source vertex to a specific vertex in the constraint graph gives the minimal x-coordinate associated to that vertex
- Hence by taking longest path to vi, it is ensured that all the inequalities in which xi participates are satisfied.
- Thus computing the longest paths to all the vertices in the constraint graph results in a solution of 1-dimensional compaction problem

Longest path algorithm for DAG

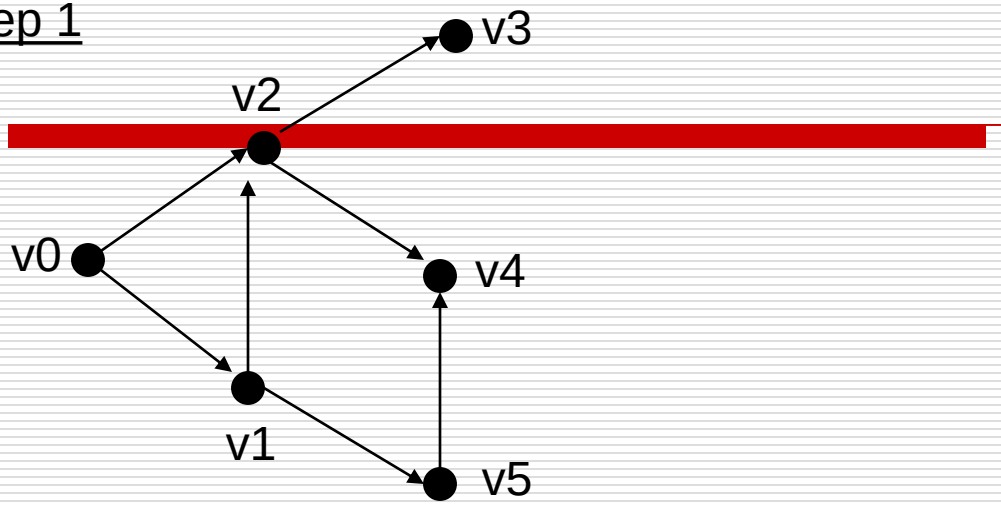
```
Longest path (G)
{
        for (i = 1; i<=n; i = i+1)
                Pi <= "in degree of vi;
        Q ← {v0}
        While (Q not equal to 0){
                vi ← "any elements from Q"
                Q ← Q / {vi}
                For each vj such that (vi, vj) ∈ E {
                    Xj ← max (xj,xi+dij)
                      Pj ← pj-1
                    If (pj <= 0)
                        Q ← Q U {vj}
                }
        }
}
```

```
main ()
{
for(i ←0; i ←n; i++)
xi ← 0;
longest-path (G);
}
```



Longest path algorithm for DAG

Step 1

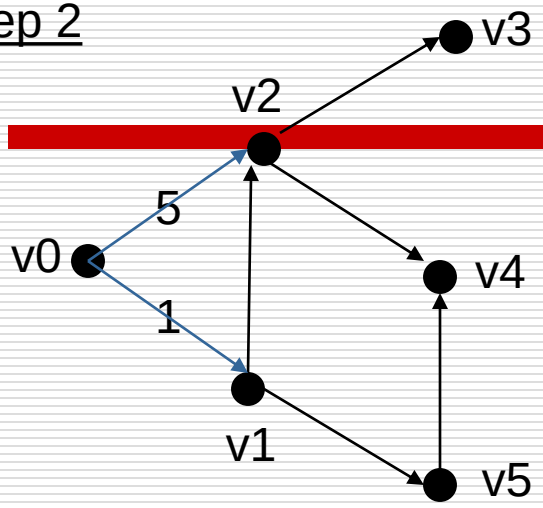


Pi ← in degree of vi

Q	P_1	P_2	P_3	P_4	P_5	X_1	X_2	X_3	X_4	X_5
initial	1	2	1	2	1	0	0	0	0	0

Longest path algorithm for DAG

Step 2



The outgoing edges from $\mathbf{V_0}$ connect $\mathbf{V_1}$ & $\mathbf{V_2}$

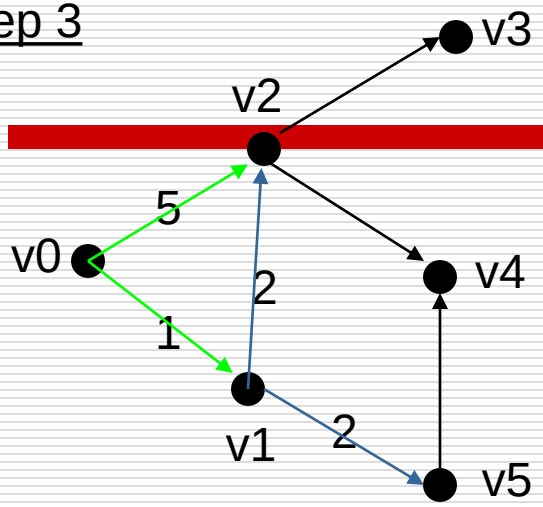
$$V_1 - V_0 \geq d_{01} \quad (1)$$

$$V_2 - V_0 \geq d_{02} \quad (5)$$

Q	P_1	P_2	P_3	P_4	P_5	X_1	X_2	X_3	X_4	X_5
initial	1	2	1	2	1	0	0	0	0	0
$\{v_0\}$	0	1	1	2	1	1	5	0	0	0

Longest path algorithm for DAG

Step 3



The outgoing edges from V_1 connect V_2 & V_5

$$V_2 - V_1 \geq d_{12} \quad (2)$$

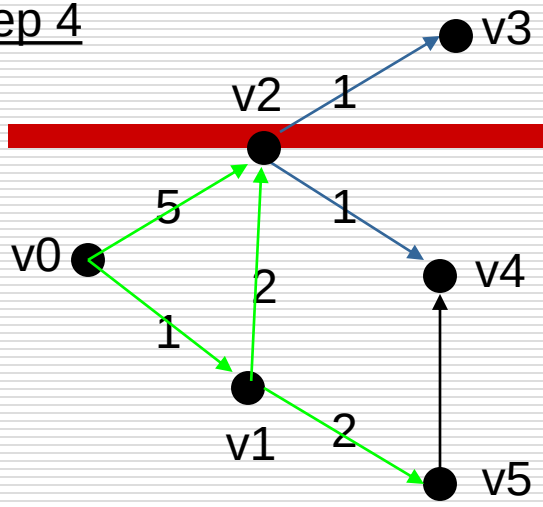
$$V_5 - V_1 \geq d_{15} \quad (2)$$

$$V_5 - V_0 \geq d_{05} \quad (1 + 2)$$

Q	P_1	P_2	P_3	P_4	P_5	X_1	X_2	X_3	X_4	X_5
initial	1	2	1	2	1	0	0	0	0	0
$\{v_0\}$	0	1	1	2	1	1	5	0	0	0
$\{v_1\}$	0	0	1	2	0	1	5	0	0	3

Longest path algorithm for DAG

Step 4



The outgoing edges from V_2 connect V_3 & V_4

$V_3 - V_2 \geq d_{23}$ (1)

$V_4 - V_2 \geq d_{24}$ (1)

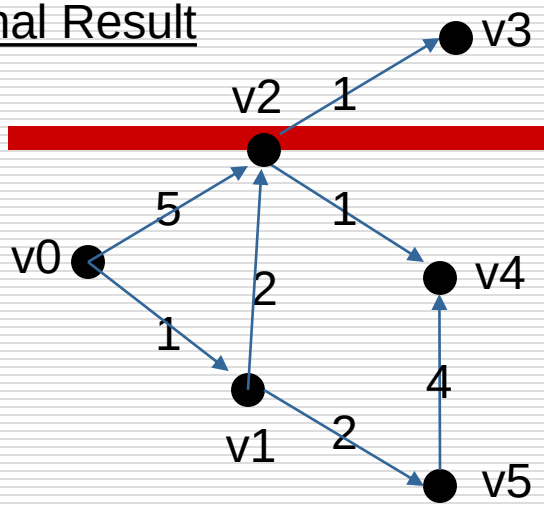
$V_3 - V_0 \geq d_{03}$ (5 + 1)

$V_4 - V_0 \geq d_{04}$ (5 + 1)

Q	P_1	P_2	P_3	P_4	P_5	X_1	X_2	X_3	X_4	X_5
initial	1	2	1	2	1	0	0	0	0	0
$\{v_0\}$	0	1	1	2	1	1	5	0	0	0
$\{v_1\}$	0	0	1	2	0	1	5	0	0	3
$\{v_2\}$	0	0	0	1	0	1	5	6	6	3

Longest path algorithm for DAG

Final Result



Q	P_1	P_2	P_3	P_4	P_5	X_1	X_2	X_3	X_4	X_5
initial	1	2	1	2	1	0	0	0	0	0
$\{v_0\}$	0	1	1	2	1	1	5	0	0	0
$\{v_1\}$	0	0	1	2	0	1	5	0	0	3
$\{v_2\}$	0	0	0	1	0	1	5	6	6	3
$\{v_3\}$	0	0	0	1	0	1	5	6	6	3
$\{v_4\}$	0	0	0	1	0	1	5	6	6	3
$\{v_5\}$	0	0	0	0	0	1	5	6	7	3