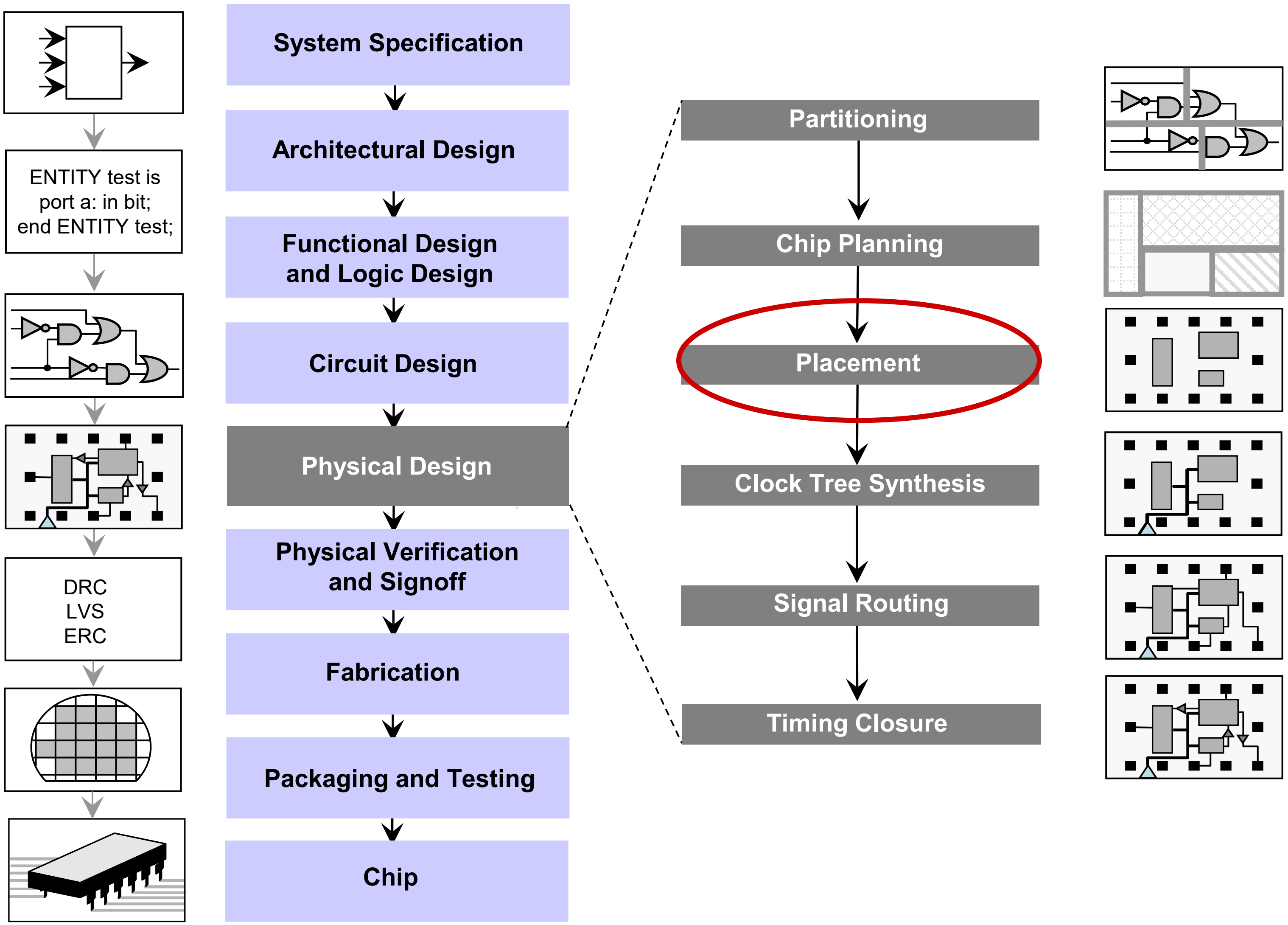
Lecture 9: Placement

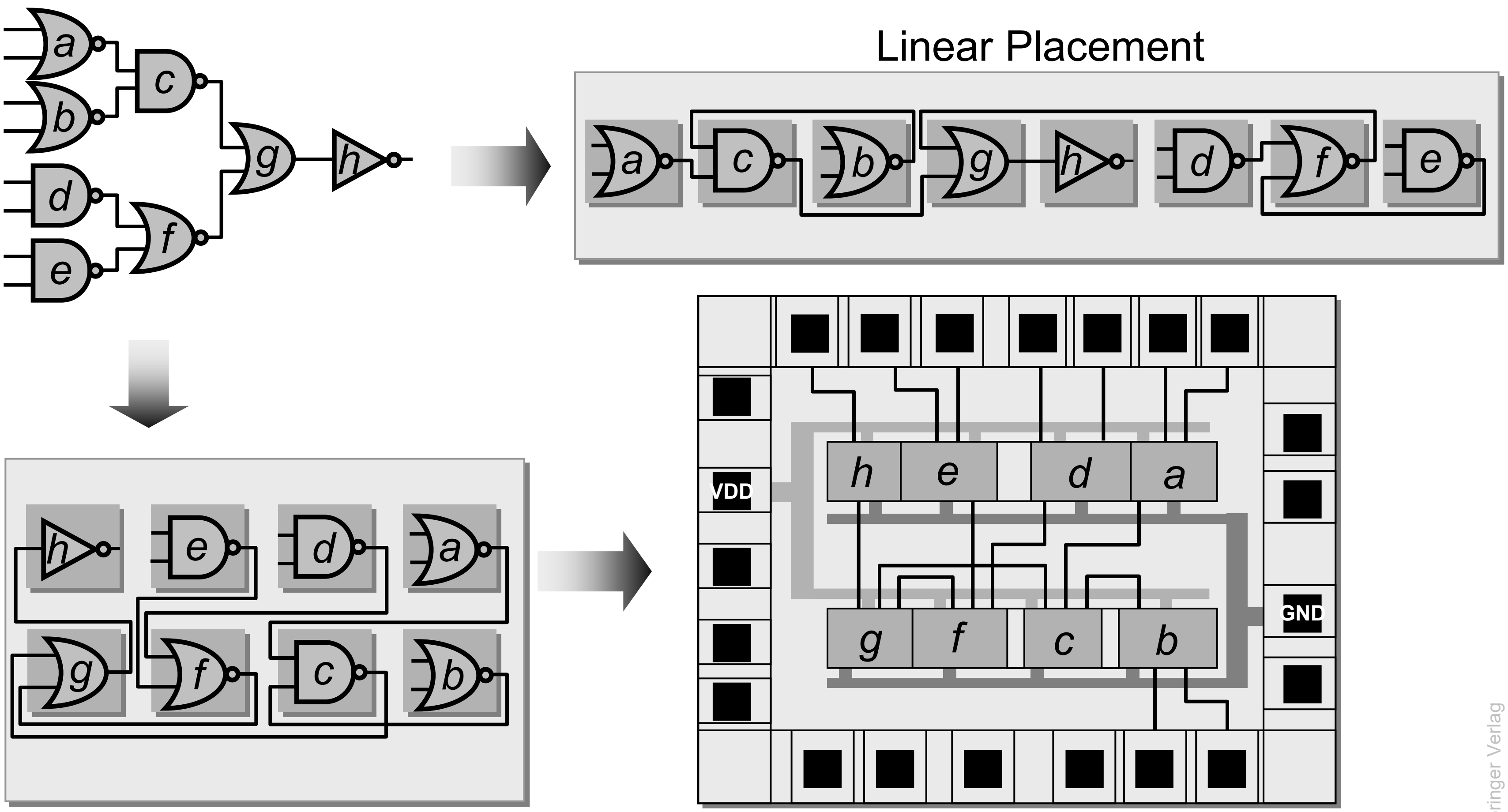
ECE201A

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

Introduction



Placement



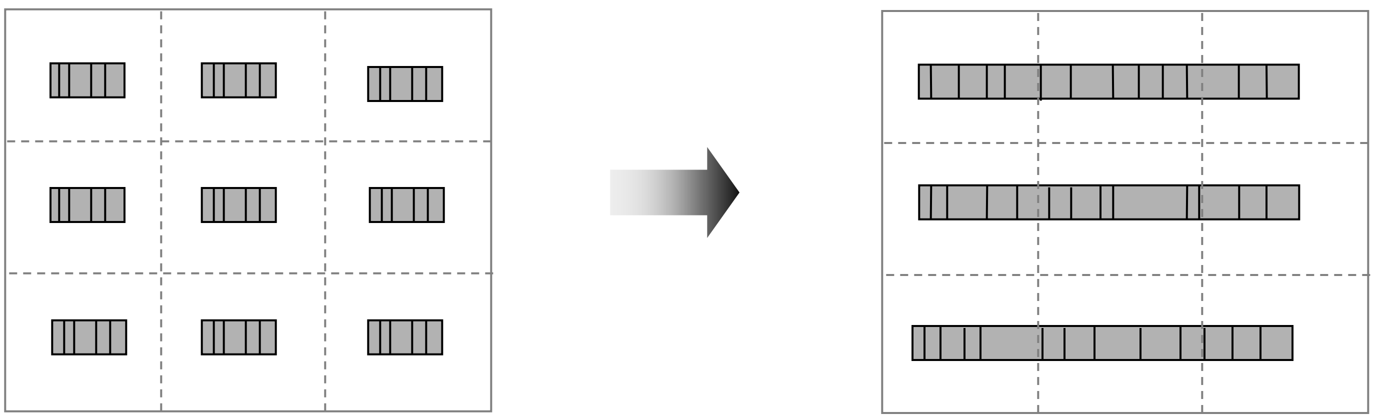
2D Placement

Placement and Routing with Standard Cells

Introduction

Global Placement

Detailed Placement



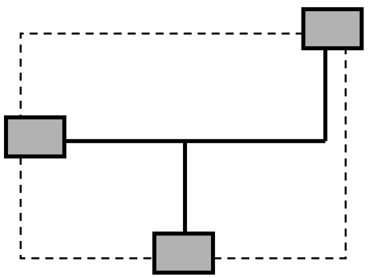
Optimization Objectives

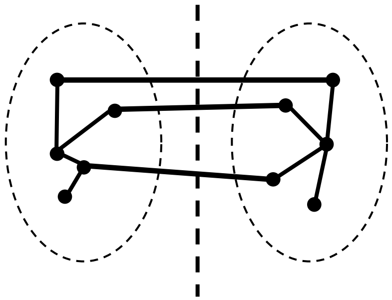
Total Wirelength

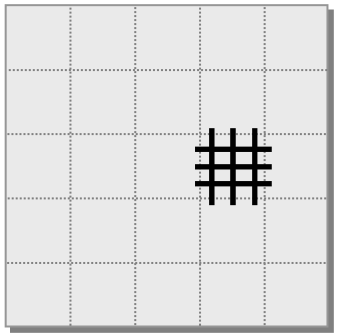
Number of Cut Nets

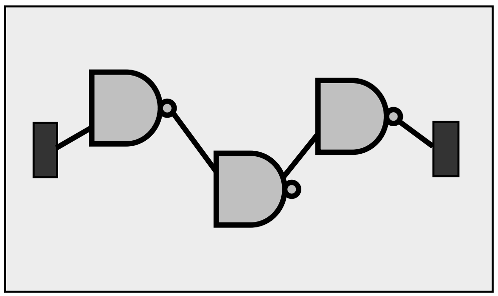
Wire Congestion

Signal Delay

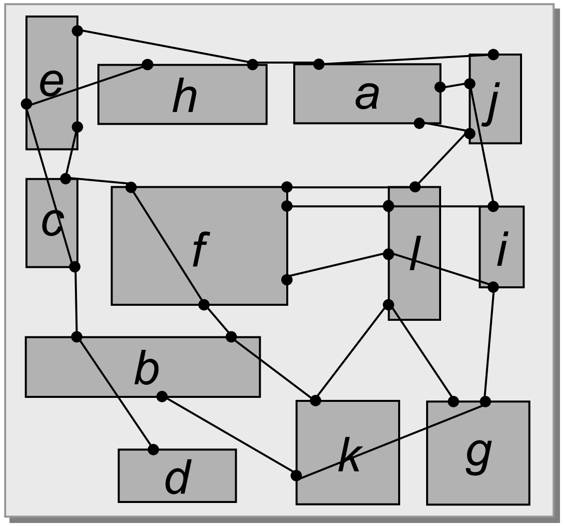


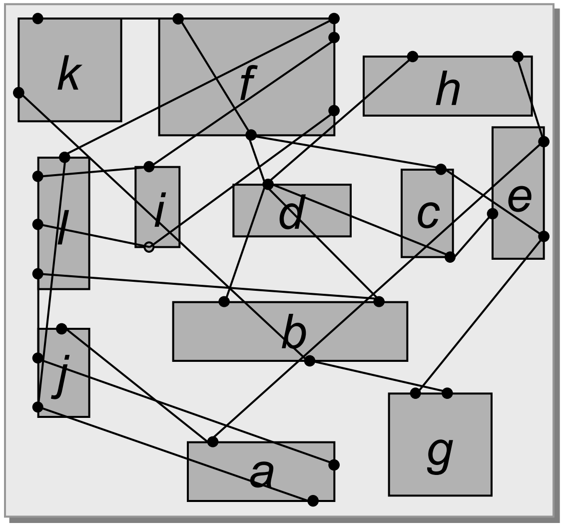




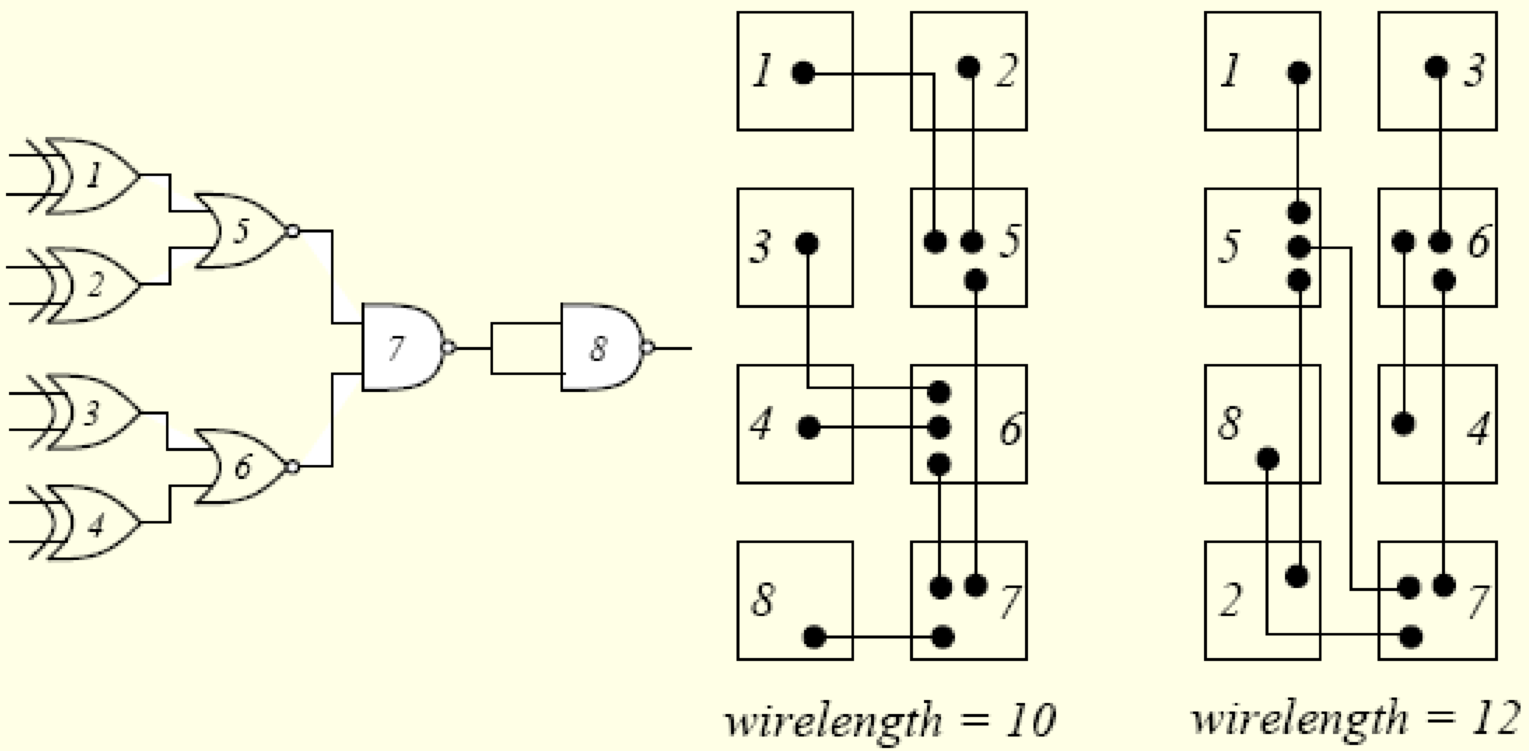


Optimization Objectives – Total Wirelength





Wirelength = Manhattan (L_1) Metric



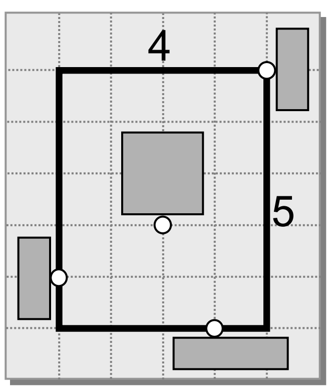
L_p norm: $((\Delta x)^p + (\Delta y)^p)^{1/p}$

p = 1 (Manhattan); p = 2 (Euclidean);
p = ∞ (max, Chebyshev)

How to measure WL ?

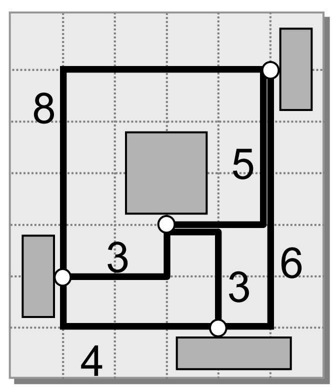
Wirelength estimation for a given placement

Half-perimeter wirelength (HPWL)



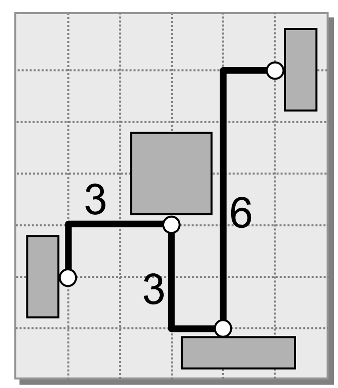
HPWL = 9
Exact for #pins = 2, 3

Complete graph (clique)



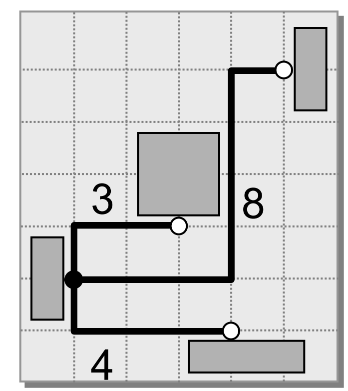
Clique Length = $(2/p)\Sigma_{e \in \text{clique}} d_M(e)$ = 14.5

Monotone chain



Chain Length = 12

Star model



Star Length = 15

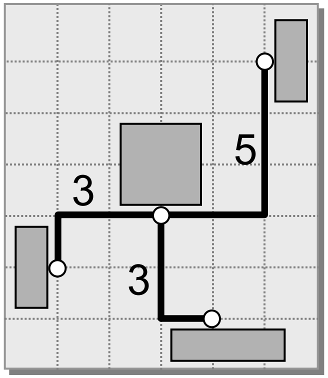
How to measure WL ?

Wirelength estimation for a given placement (cont'd.)

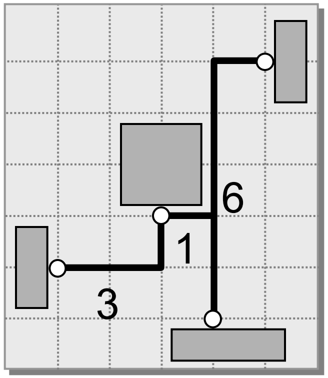
Rectilinear minimum spanning tree (RMST)

Rectilinear Steiner minimum tree (RSMT)

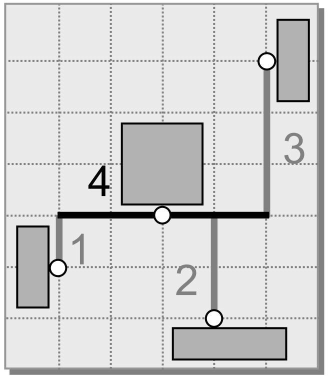
Single-trunk Steiner tree (STST)



RMST Length = 11
O(n log n)



RSMT Length = 10
NP-hard optimization
True WL



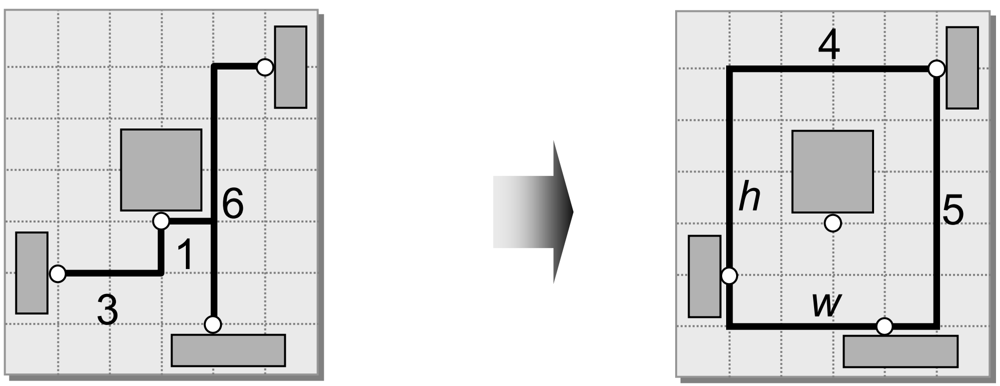
STST Length = 10

Optimization Objectives – Total Wirelength

Wirelength estimation for a given placement (cont'd.)

Preferred method: Half-perimeter wirelength (HPWL)

- Fast (order of magnitude faster than RSMT)
- Equal to length of RSMT for 2- and 3-pin nets
- Margin of error for real circuits approx. 8%



$$L_{\mathrm{HPWL}} = w + h$$

RSMT Length = 10

HPWL = 9

Optimization Objectives – Total Wirelength

Total wirelength with net weights (weighted wirelength)

- For a placement P, an estimate of total weighted wirelength is

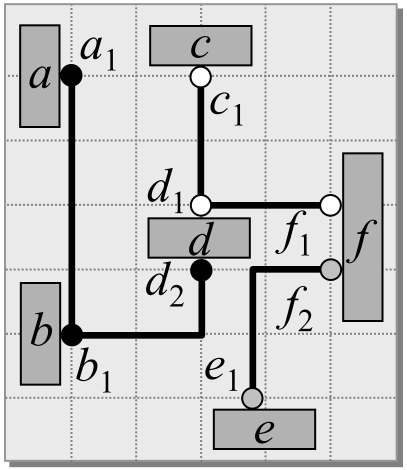
$$L(P) = \sum_{net \in P} w(net) \cdot L(net)$$

where $w(net)$ is the weight of net, and $L(net)$ is the estimated wirelength of net.

Weight could be timing criticality

- Example:

Nets	Weights
$N_1 = (a_1, b_1, d_2)$	$w(N_1) = 2$
$N_2 = (c_1, d_1, f_1)$	$w(N_2) = 4$
$N_3 = (e_1, f_2)$	$w(N_3) = 1$



$$L(P) = \sum_{net \in P} w(net) \cdot L(net) = 2 \cdot 7 + 4 \cdot 4 + 1 \cdot 3 = 33$$

Optimization Objectives – Number of Cut Nets

Cut sizes of a placement

- To improve total wirelength of a placement *P*, separately calculate the number of crossings of global vertical and horizontal cutlines, and minimize

$$L(P) = \sum_{v \in V_P} \psi_P(v) + \sum_{h \in H_P} \psi_P(h)$$

where $\Psi_P(cut)$ be the set of nets cut by a cutline *cut*

- Net-cut cost = number of external nets between different global bins
  - Asymptotically, net-cut cost = wire length
  - Minimizing net-cut in global placement tends to put highly connected cells close to each other.

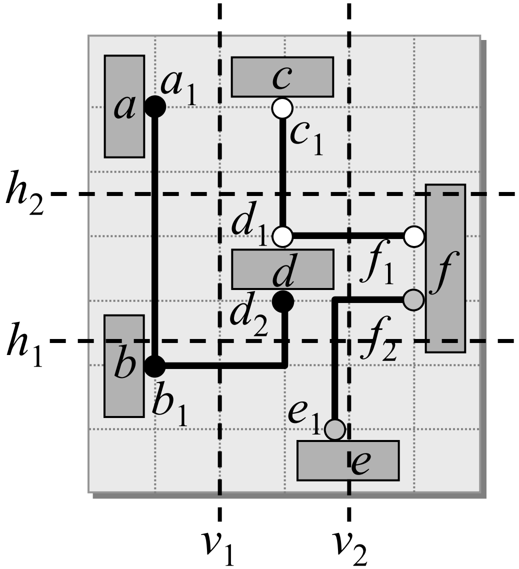
Optimization Objectives – Number of Cut Nets

Cut sizes of a placement

- Example:

  Nets

  $N_1 = (a_1, b_1, d_2)$
  $N_2 = (c_1, d_1, f_1)$
  $N_3 = (e_1, f_2)$



- Cut values for each global cutline

  $\psi_P(v_1) = 1 \quad \psi_P(v_2) = 2$

  $\psi_P(h_1) = 3 \quad \psi_P(h_2) = 2$

- Total number of crossings in P

  $\psi_P(v_1) + \psi_P(v_2) + \psi_P(h_1) + \psi_P(h_2) = 1 + 2 + 3 + 2 = 8$

- Cut sizes

  $X(P) = \max(\psi_P(v_1),\psi_P(v_2)) = \max(1,2) = 2$

  $Y(P) = \max(\psi_P(h_1),\psi_P(h_2)) = \max(3,2) = 3$

Optimization Objectives – Wire Congestion

Routing congestion of a placement

- Formally, the local wire density $\varphi_P(e)$ of an edge e between two neighboring grid cells is

$$\varphi_P(e) = \frac{\eta_P(e)}{\sigma_P(e)}$$

where $\eta_P(e)$ is the estimated number of nets that cross e and
$\sigma_P(e)$ is the maximum number of nets that can cross e

- If $\varphi_P(e) > 1$, then too many nets are estimated to cross e, making P more likely to be unroutable.
- The wire density of P is $\Phi(P) = \max_{e \in E}\left(\varphi_P(e)\right)$

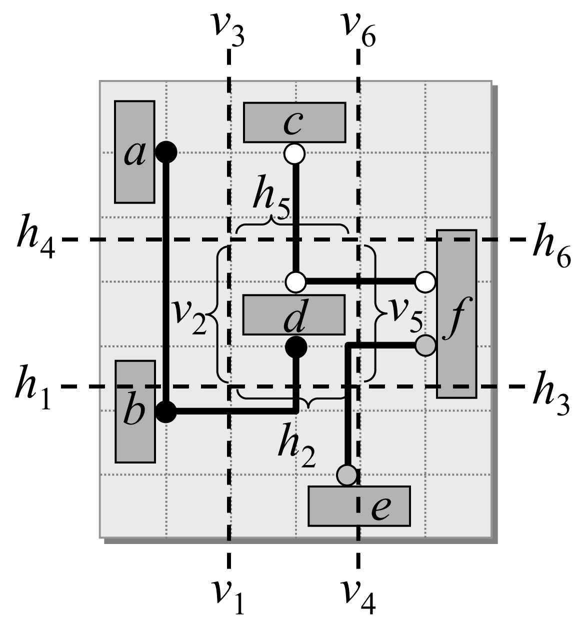
where E is the set of all edges

- If $\Phi(P) < 1$, then the design is estimated to be fully routable, otherwise routing will need to detour some nets through less-congested edges

Optimization Objectives – Wire Congestion

Wire Density of a placement

$\eta_P(h_1) = 1$ $\quad$ $\eta_P(v_1) = 1$
$\eta_P(h_2) = 2$ $\quad$ $\eta_P(v_2) = 0$
$\eta_P(h_3) = 0$ $\quad$ $\eta_P(v_3) = 0$
$\eta_P(h_4) = 1$ $\quad$ $\eta_P(v_4) = 0$
$\eta_P(h_5) = 1$ $\quad$ $\eta_P(v_5) = 2$
$\eta_P(h_6) = 0$ $\quad$ $\eta_P(v_6) = 0$



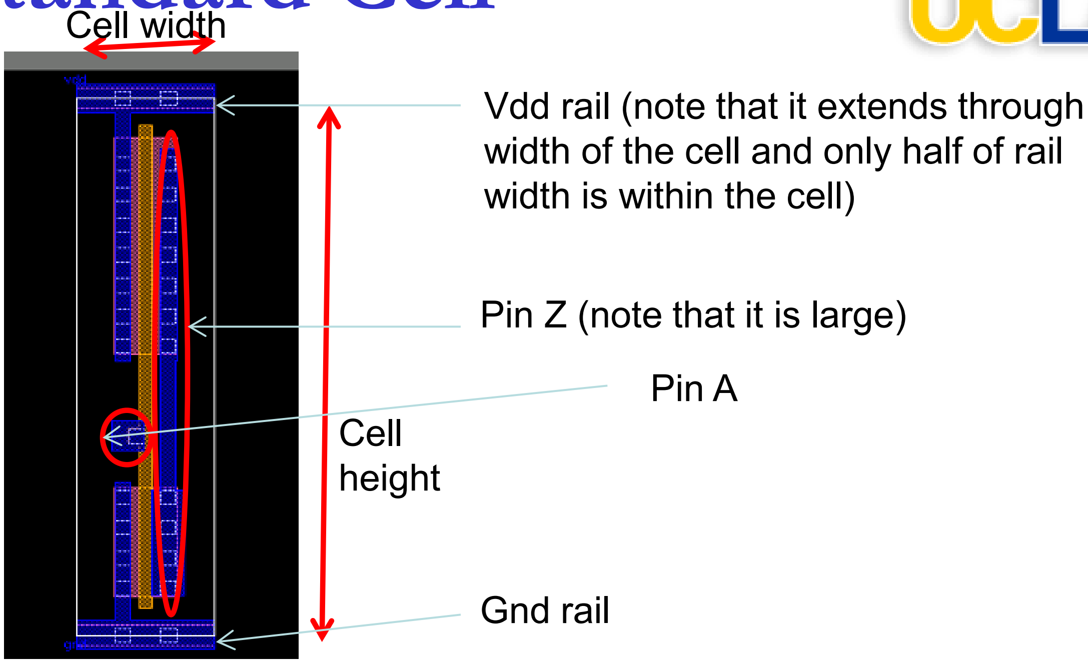
Maximum: $\eta_P(e) = 2$

$$\boxed{\Phi(P) = \frac{\eta_P(e)}{\sigma_P(e)} = \frac{2}{3}}$$

Routable

The Standard Cell

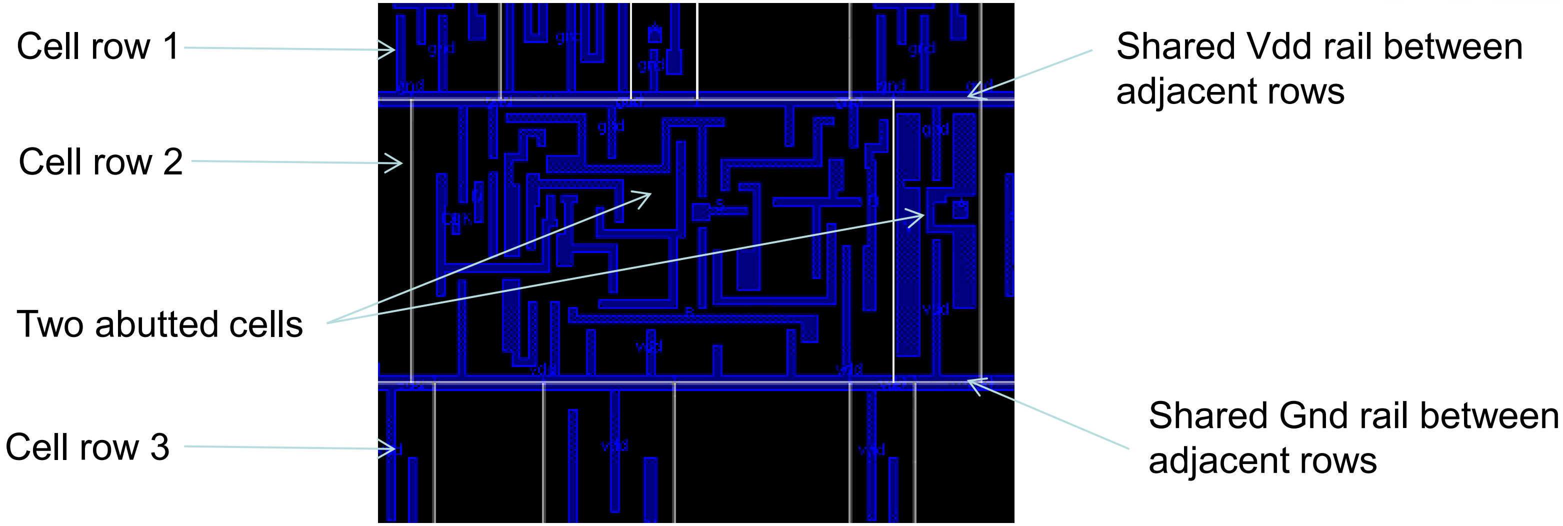
Cell layouts are front-end only: typically use only up to M1 (sometimes M2)



An INV Cell

- Almost all modern digital designs are cell-based.
  - Cells are typically small gates (NAND, FF, etc) containing 2-100 transistors.
- There are ~1000 cells in a library
  - Typically one design uses one library only.
- Cell width and height are quantized (measured in #metal tracks)
  - Cell height usually fixed for a library (8-track, 10-track, etc)
  - Cell width a multiple of minimum sized inverter (again multiple of min. metal pitch)
  - Constraint ensures that when cells are placed abutted, all pins fall on metal track intersections → easier for router to connect to them.

Cell-Based Placement



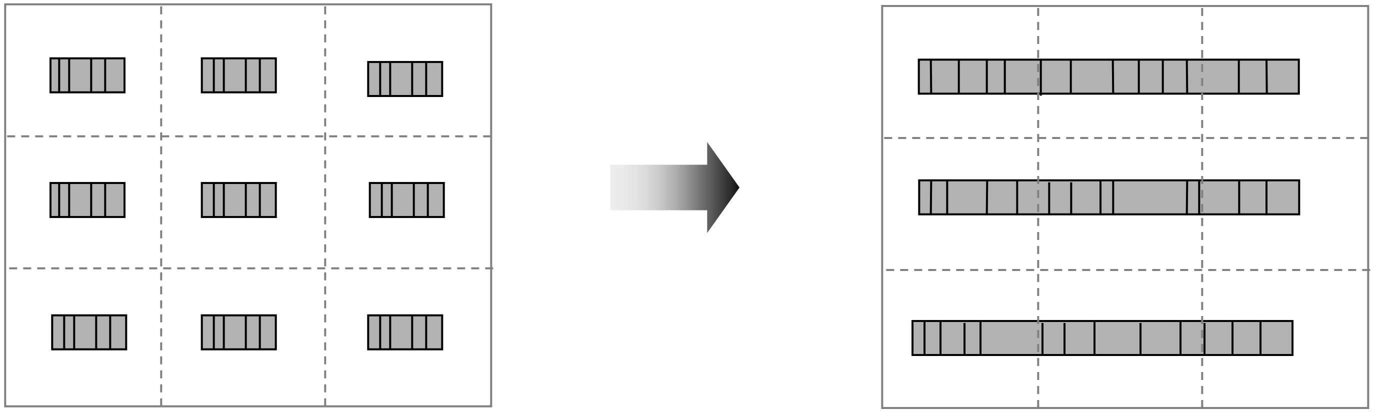
- Cells are placed in rows
- Almost all logic is automatically synthesized, placed using cell libraries
- Modern designs easily have 10M+ cells
- Goal of placement: timing correct, routable placement of all cells

5 min break

Introduction

Global Placement

Detailed Placement



Global Placement

- Partitioning-based algorithms:
  - The netlist and the layout are divided into smaller sub-netlists and sub-regions, respectively
  - Process is repeated until each sub-netlist and sub-region is small enough to be handled optimally
  - Detailed placement often performed by optimal solvers, facilitating a natural transition from global placement to detailed placement
  - Example: min-cut placement
- Analytic techniques:
  - Model the placement problem using an objective (cost) function, which can be optimized via numerical analysis
  - Examples: quadratic placement and force-directed placement
- Stochastic algorithms:
  - Randomized moves that allow hill-climbing are used to optimize the cost function
  - Example: simulated annealing

Global Placement

Partitioning-based

Analytic

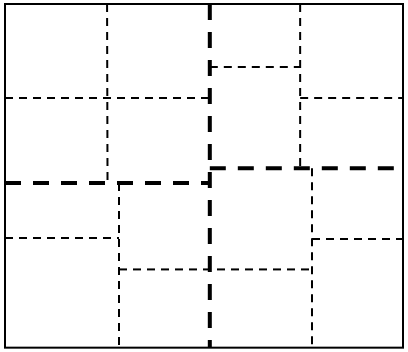
Stochastic

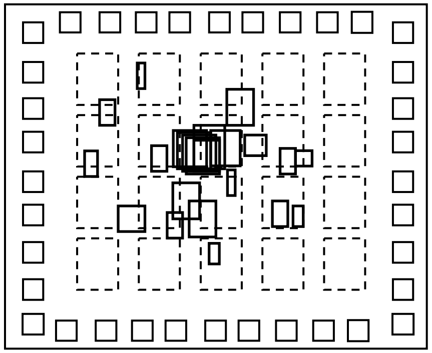
Min-cut placement

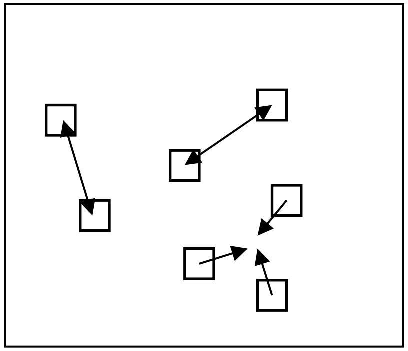
Quadratic placement

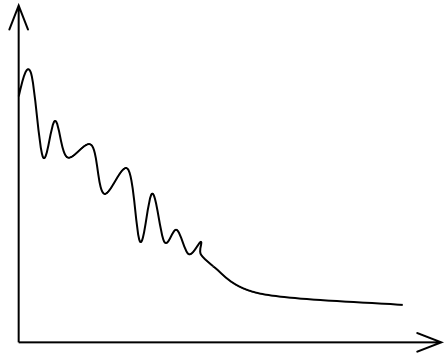
Force-directed placement

Simulated annealing

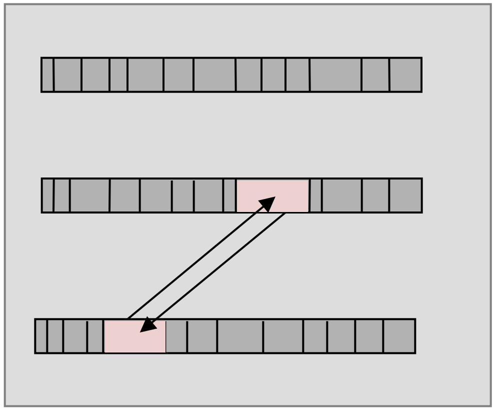


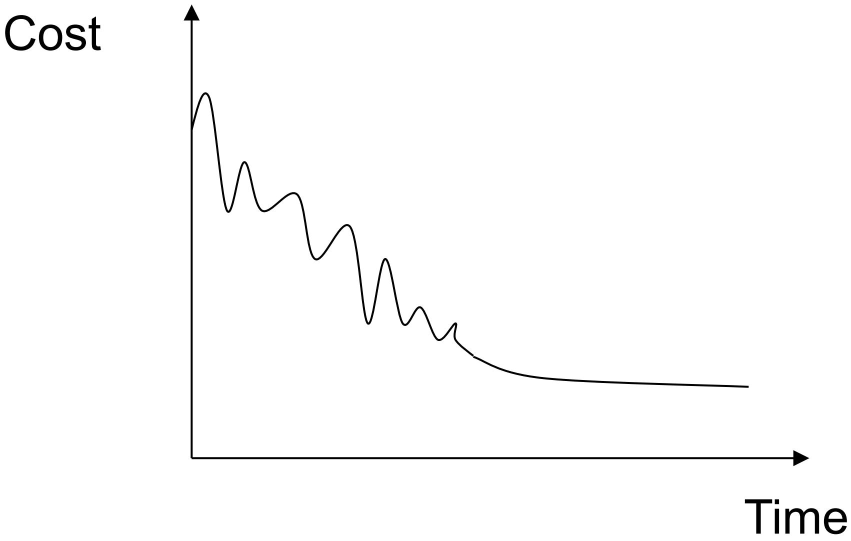






Simulated Annealing

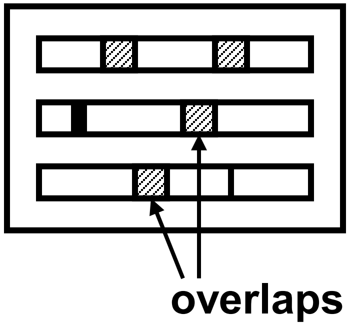




- Analogous to the physical annealing process
  - Melt metal and then slowly cool it
  - Result: energy-minimal crystal structure
- Modification of an initial configuration (placement) by moving/exchanging of randomly selected cells
  - Accept the new placement if it improves the objective function
  - If no improvement: Move/exchange is accepted with temperature-dependent (i.e., decreasing) probability

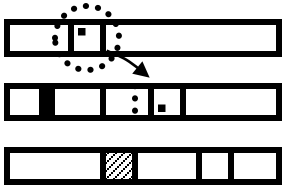
Neighborhood Structure for SA

Solution space: All arrangements of cells into rows, possibly with overlaps

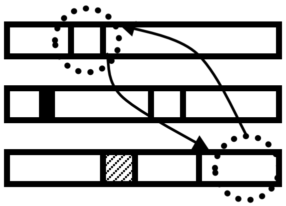


Three types of moves:

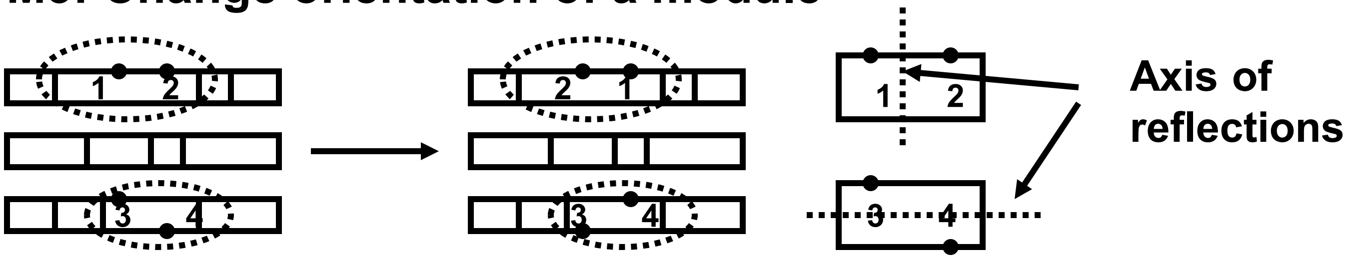
M1: Displace a module to a new location



M2: Interchange two modules



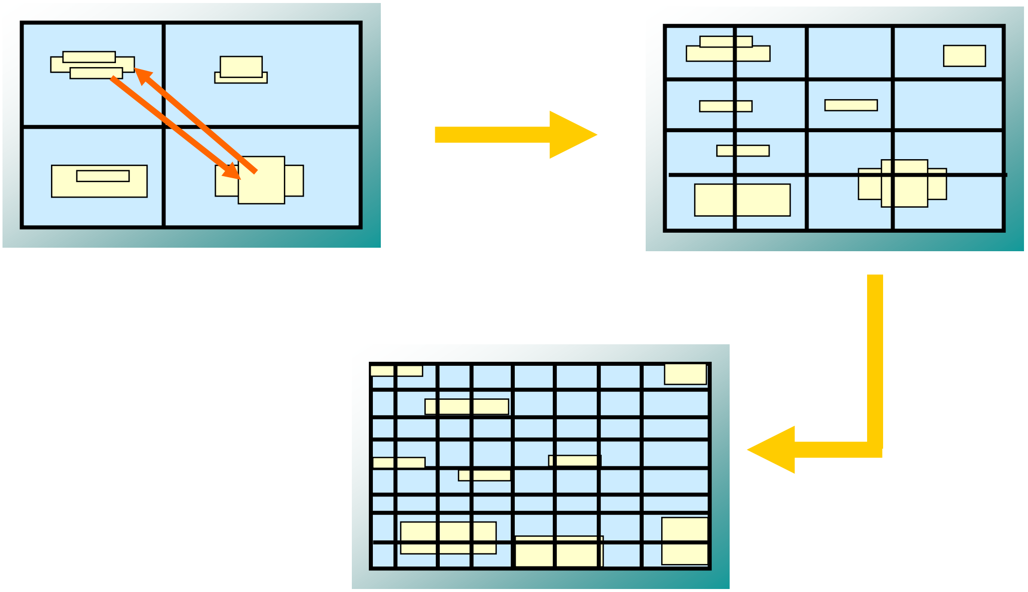
M3: Change orientation of a module



Simulated Annealing

- Advantages:
  - Can find global optimum (given sufficient time)
  - Well-suited for detailed placement
- Disadvantages:
  - Very slow
  - To achieve high-quality implementation, laborious parameter tuning is necessary
  - Randomized, chaotic algorithms - small changes in the input lead to large changes in the output
- Practical applications of SA:
  - Very small placement instances with complicated constraints
  - Detailed placement, where SA can be applied in small windows (not common anymore)
  - FPGA layout, where complicated constraints are becoming a norm

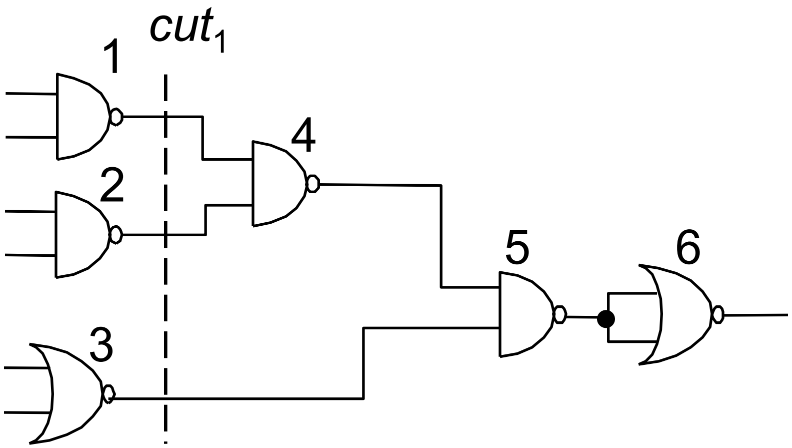
Min-Cut Placement



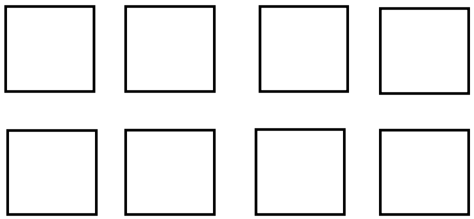
- Uses partitioning algorithms to divide (1) the netlist and (2) the layout region into smaller sub-netlists and sub-regions
- Conceptually, each sub-region is assigned a portion of the original netlist
- Each cut heuristically minimizes the number of cut nets using,
  - Fiduccia-Mattheyses (FM) algorithm

Min-Cut Placement – Example

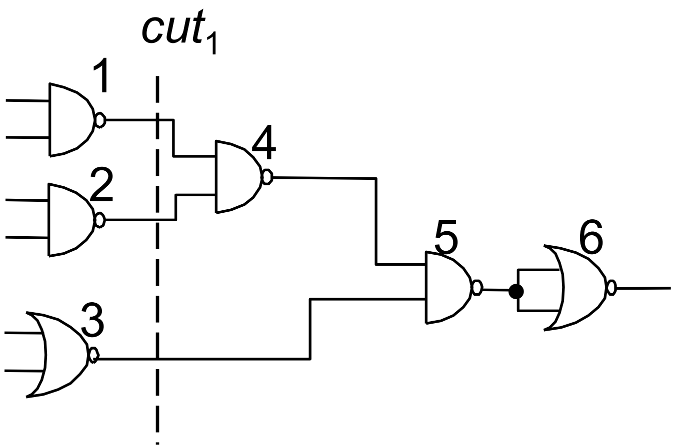
Given:



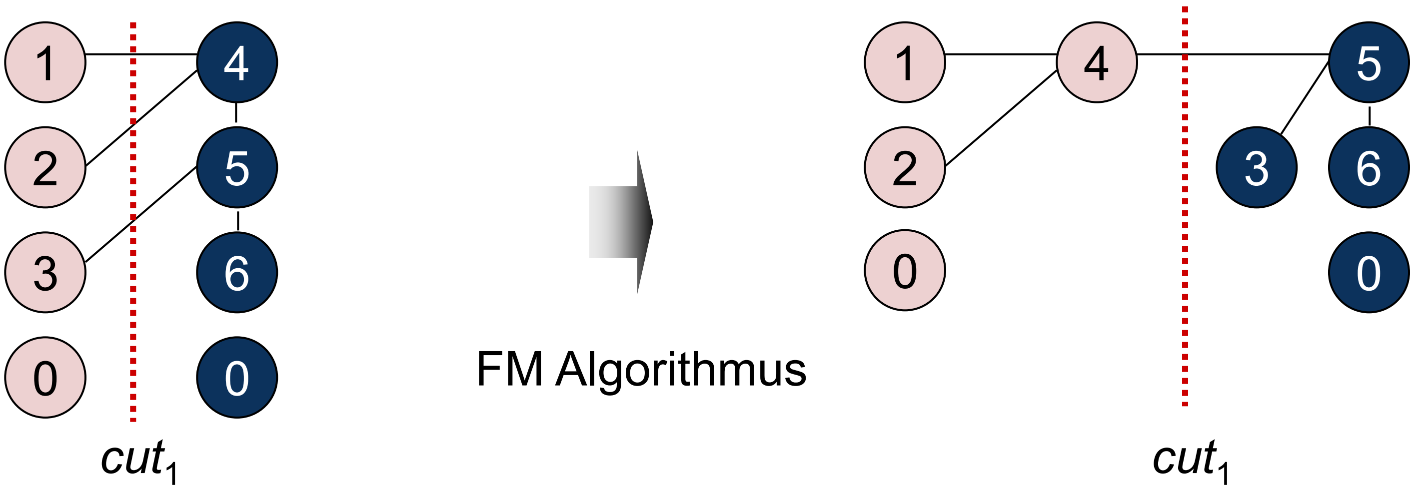
Task: 4 x 2 placement with minimum wirelength using alternative cutline directions and the FM algorithm

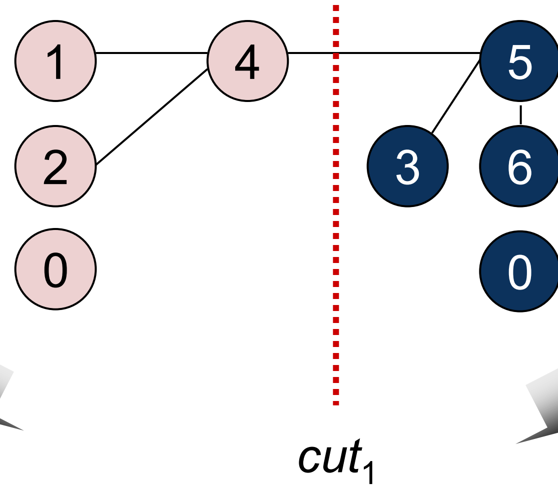


Min-Cut-Placement



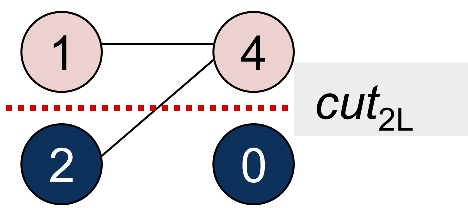
Vertical cut cut_1: L={1,2,3}, R={4,5,6}

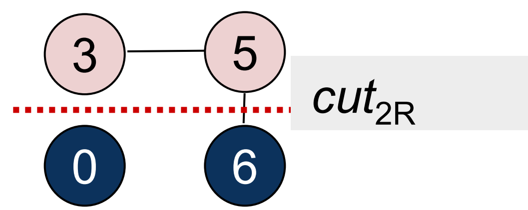


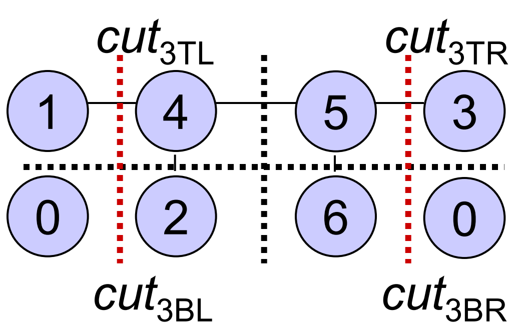


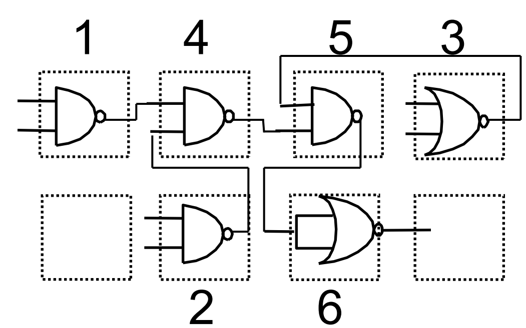
Horizontal cut cut_{2L}: T={1,4}, B={2,0}

Horizontal cut cut_{2R}: T={3,5}, B={6,0}

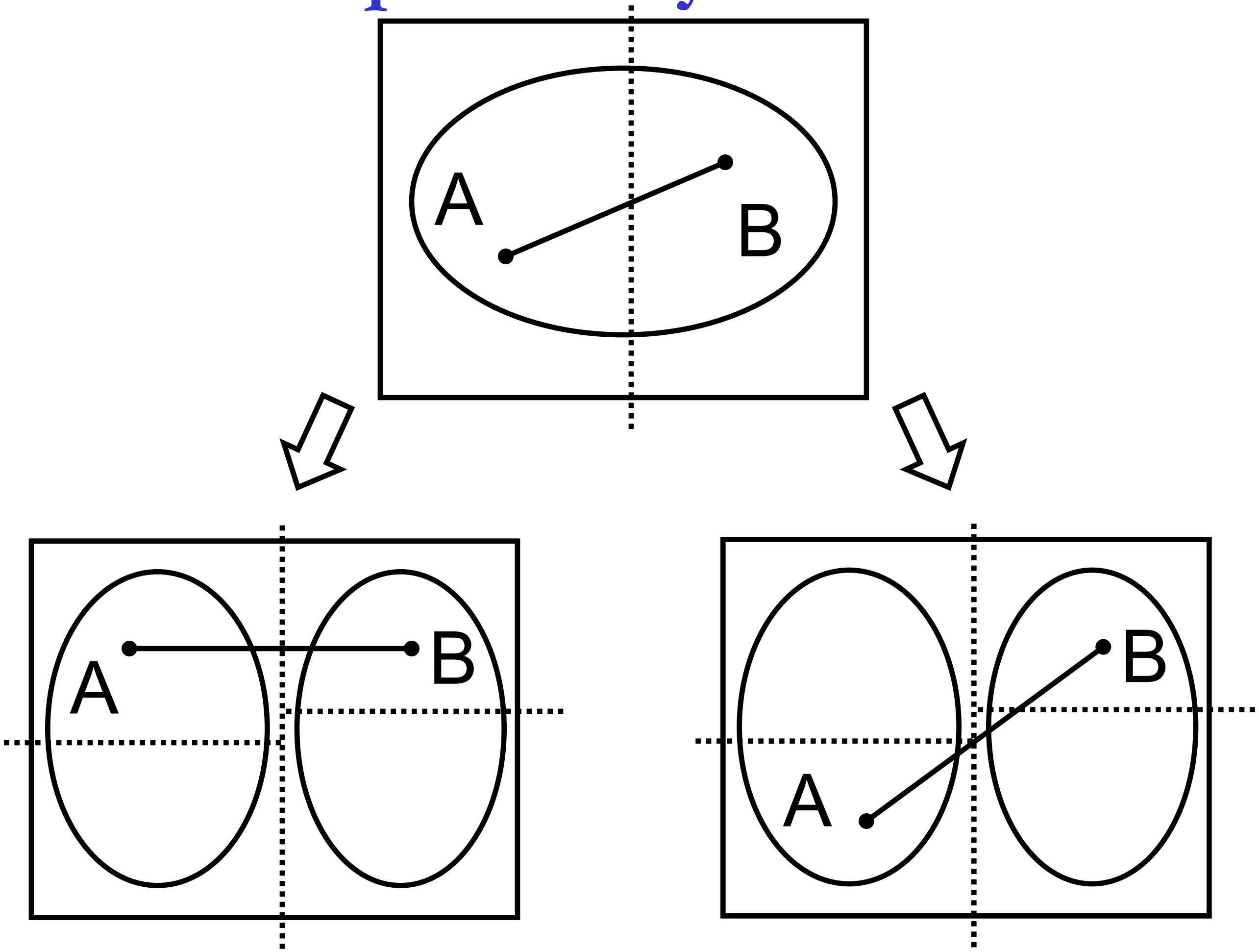






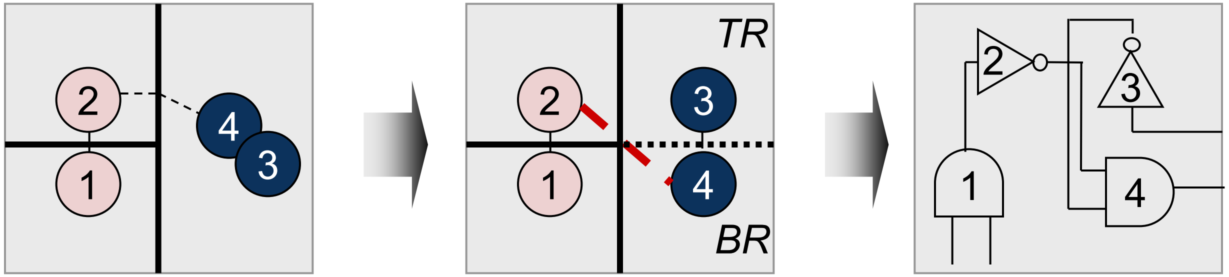


How to Partition Subcircuits "Independently"?

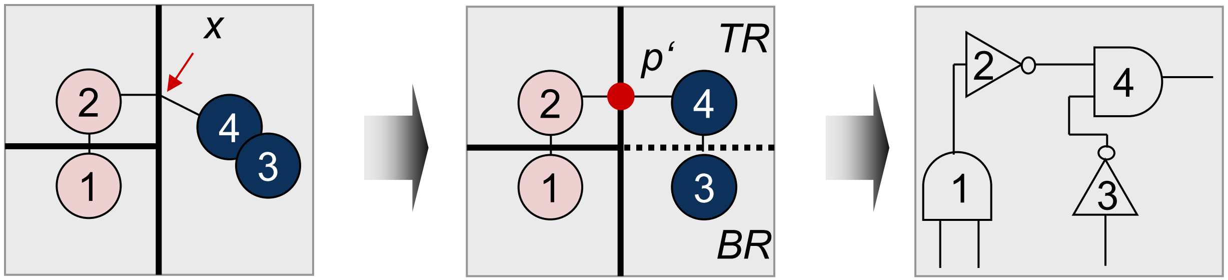


The costs of these two solutions are not the same

Min-Cut Placement – Terminal Propagation



- Terminal Propagation
  - External connections are represented by artificial connection points on the cutline
  - Dummy nodes in hypergraphs



Min-Cut Placement Summary

- Advantages:
  - Reasonable fast
  - Objective function can be adjusted, e.g., to perform timing-driven placement
  - Hierarchical strategy applicable to large circuits

- Disadvantages:
  - Randomized, chaotic algorithms – small changes in input lead to large changes in output
  - Optimizing one cutline at a time may result in routing congestion elsewhere

Analytic Placement – Quadratic Placement

- Objective function is quadratic; sum of (weighted) squared Euclidean distance represents placement objective function

$$L(P) = \frac{1}{2}\sum_{i,j=1}^{n} c_{ij}\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right)$$

where *n* is the total number of cells, and *c*(*i*,*j*) is the connection cost between cells *i* and *j*.

- Only two-point-connections
- Minimize objective function by equating its derivative to zero which reduces to solving a system of linear equations

Analytic Placement – Quadratic Placement

$$L(P) = \frac{1}{2}\sum_{i,j=1}^{n} c_{ij}\left((x_i - x_j)^2 + (y_i - y_j)^2\right)$$

where n is the total number of cells, and $c(i,j)$ is the connection cost between cells i and j.

- Each dimension can be considered independently:

$$L_x(P) = \sum_{i=1,\, j=1}^{n} c(i,j)(x_i - x_j)^2 \qquad L_y(P) = \sum_{i=1,\, j=1}^{n} c(i,j)(y_i - y_j)^2$$

- Convex quadratic optimization problem: any local minimum solution is also a global minimum
- Optimal x- and y-coordinates can be found by setting the partial derivatives of $L_x(P)$ and $L_y(P)$ to zero

Analytic Placement – Quadratic Placement

$$L(P) = \frac{1}{2}\sum_{i,j=1}^{n} c_{ij}\left(\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2\right)$$

where n is the total number of cells, and $c(i,j)$ is the connection cost between cells i and j.

$$L_x(P) = \sum_{i=1, j=1}^{n} c(i,j)(x_i - x_j)^2$$

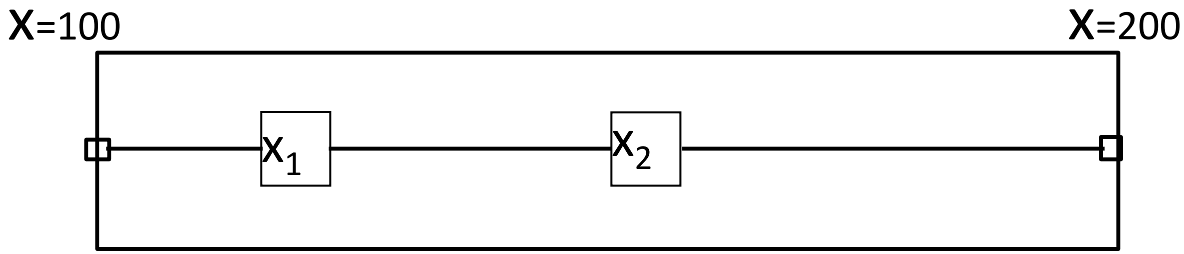
$$\frac{\partial L_x(P)}{\partial X} = AX - b_x = 0$$

$$L_y(P) = \sum_{i=1, j=1}^{n} c(i,j)(y_i - y_j)^2$$

$$\frac{\partial L_y(P)}{\partial Y} = AY - b_y = 0$$

- Each dimension can be considered independently

Analytical Placement Example



$$Cost = (x_1 - 100)^2 + (x_1 - x_2)^2 + (x_2 - 200)^2$$

$$\frac{\partial}{\partial x_1} Cost = 2(x_1 - 100) + 2(x_1 - x_2)$$

$$\frac{\partial}{\partial x_2} Cost = -2(x_1 - x_2) + 2(x_2 - 200)$$

Setting the partial derivatives $= 0$, we solve for the minimum Cost :

$$Ax = b$$

$$\begin{bmatrix} 4 & -2 \\ -2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 200 \\ 400 \end{bmatrix}$$

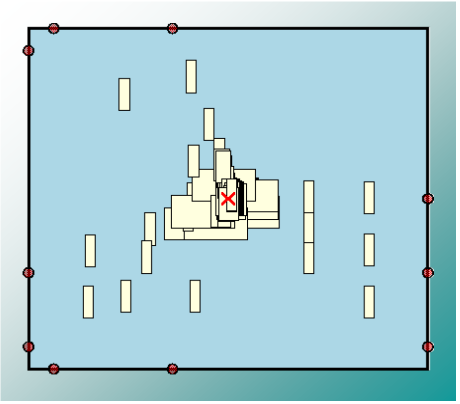
$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \end{bmatrix}$$

$$x_1 = \frac{400}{3} \qquad x_2 = \frac{500}{3}$$

A_{ii} = degree of a node
A_{ij} = -(i-j) connectivity
b_i = sum of locations connected to cell i

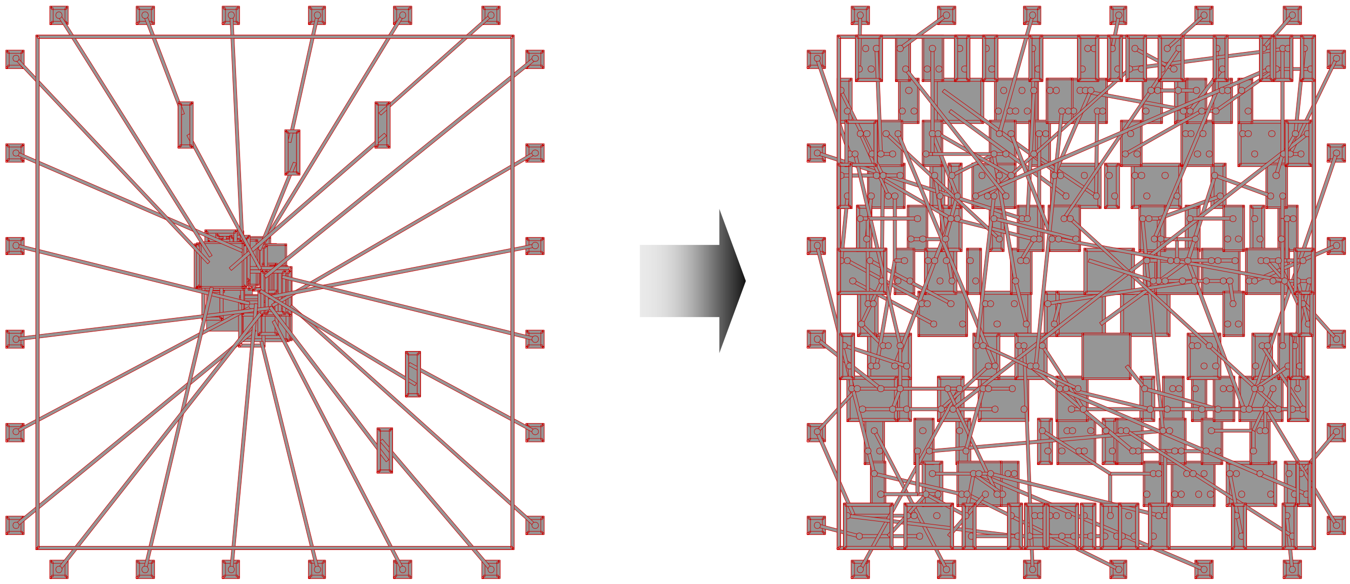
Why “Squared Wirelength” ?

- Because we can
- Because it is trivial to solve
- Because there is only one solution
- Because the solution is a global optimum
- Because the solution conveys “relative order” information
- (Because the solution conveys “global position” information)
- **Key issue: “spreading”**
  - What is the optimal Solution in previous case if No pin locations are there ?



Analytic Placement – Quadratic Placement

- Second stage of quadratic placers: cells are spread out to remove overlaps
- Methods:
  - Adding fake nets that pull cells away from dense regions toward anchors
  - Geometric sorting and scaling
  - Repulsion forces, etc.



Analytic Placement – Quadratic Placement

- Advantages:
  - Captures the placement problem concisely in mathematical terms
  - Leverages efficient algorithms from numerical analysis and available software
  - Can be applied to large circuits without netlist clustering (flat)
  - Stability: small changes in the input do not lead to large changes in the output
- Disadvantages:
  - Connections to fixed objects are necessary: I/O pads, pins of fixed macros, etc.