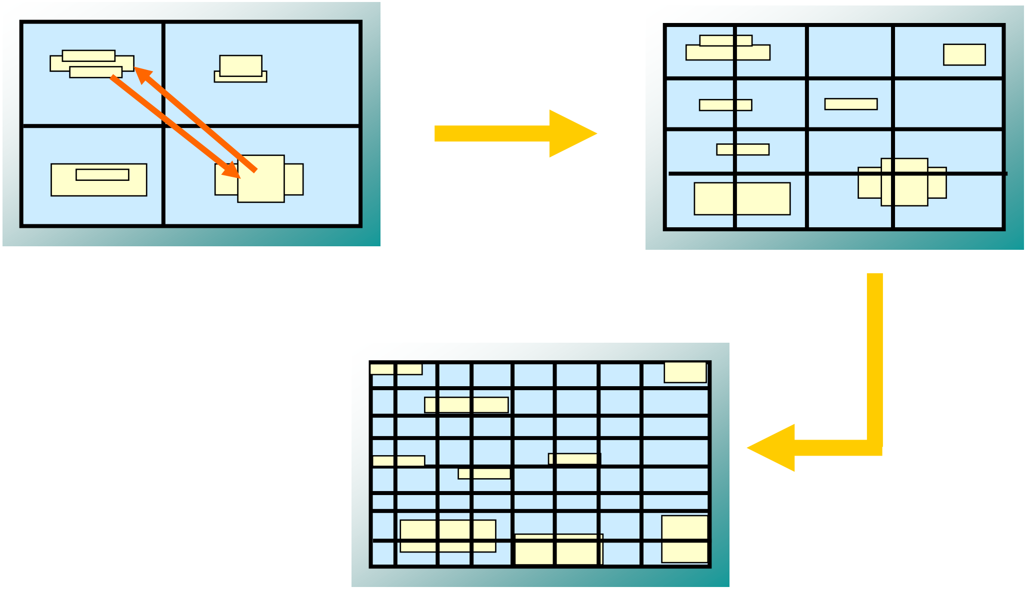
Logistics

- I hope you have formed project teams
- Make progress toward your midterm report

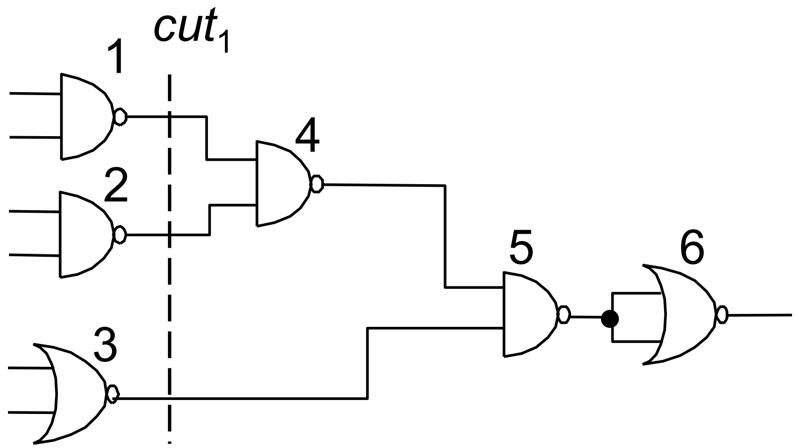
Min-Cut Placement



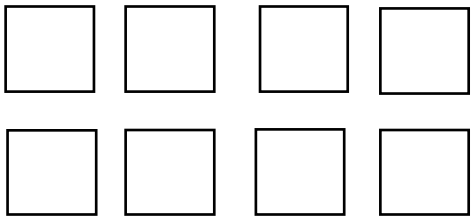
- Uses partitioning algorithms to divide (1) the netlist and (2) the layout region into smaller sub-netlists and sub-regions
- Conceptually, each sub-region is assigned a portion of the original netlist
- Each cut heuristically minimizes the number of cut nets using,
  - Fiduccia-Mattheyses (FM) algorithm

Min-Cut Placement – Example

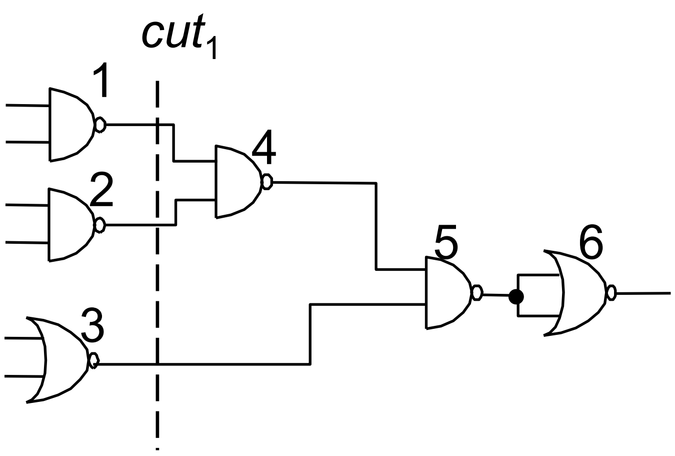
Given:



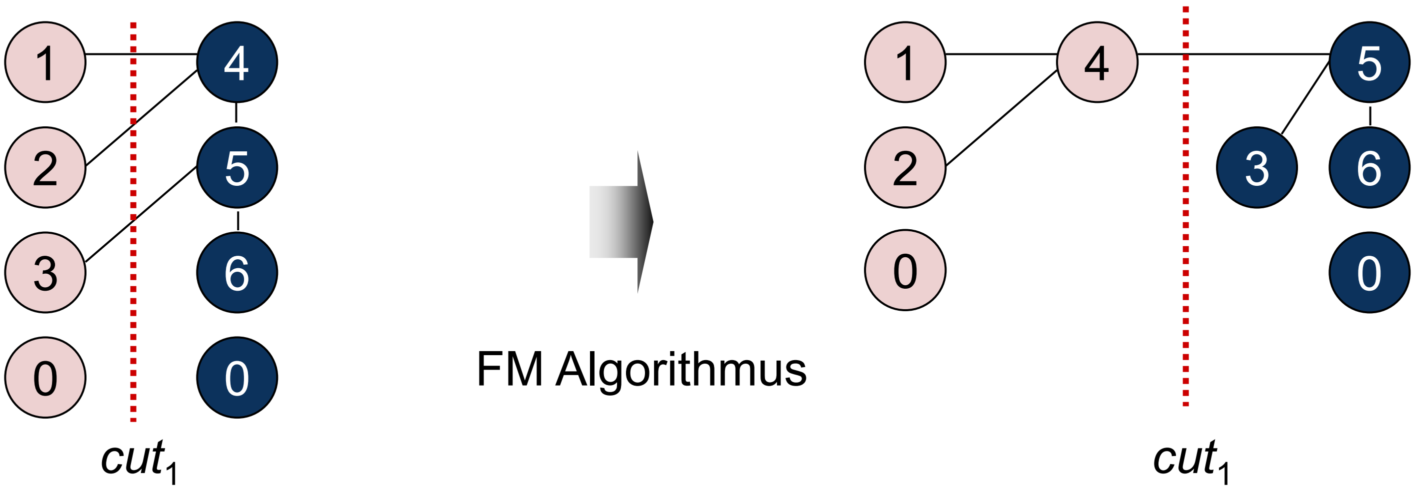
Task: 4 x 2 placement with minimum wirelength using alternative cutline directions and the FM algorithm

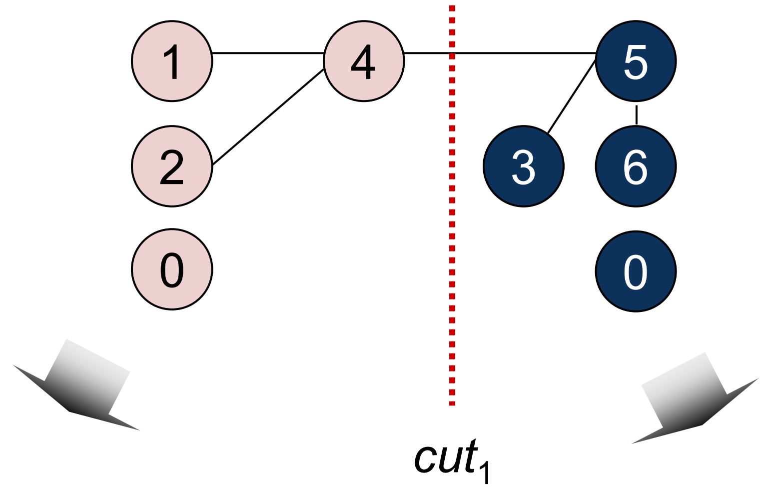


Min-Cut-Placement



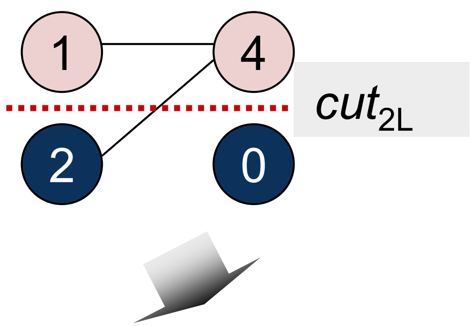
Vertical cut cut_1: L={1,2,3}, R={4,5,6}

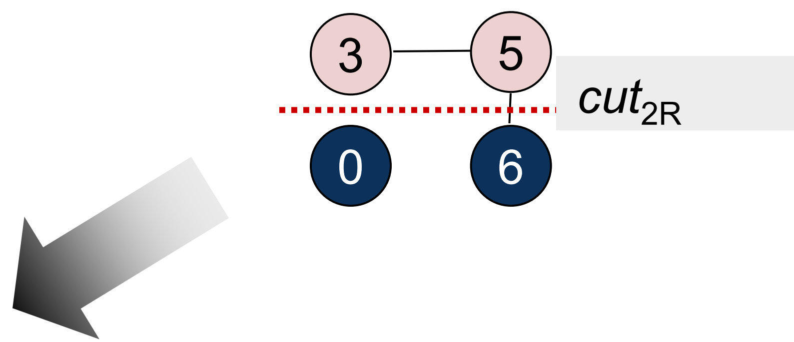


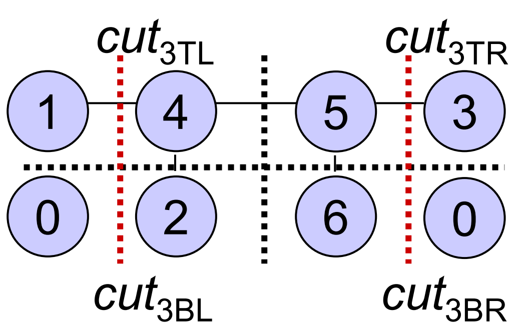


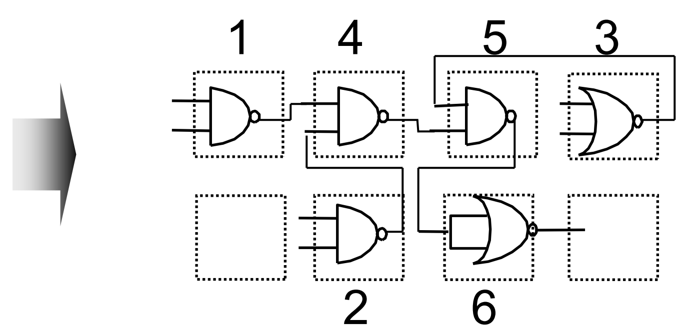
Horizontal cut cut_{2L}: T={1,4}, B={2,0}

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

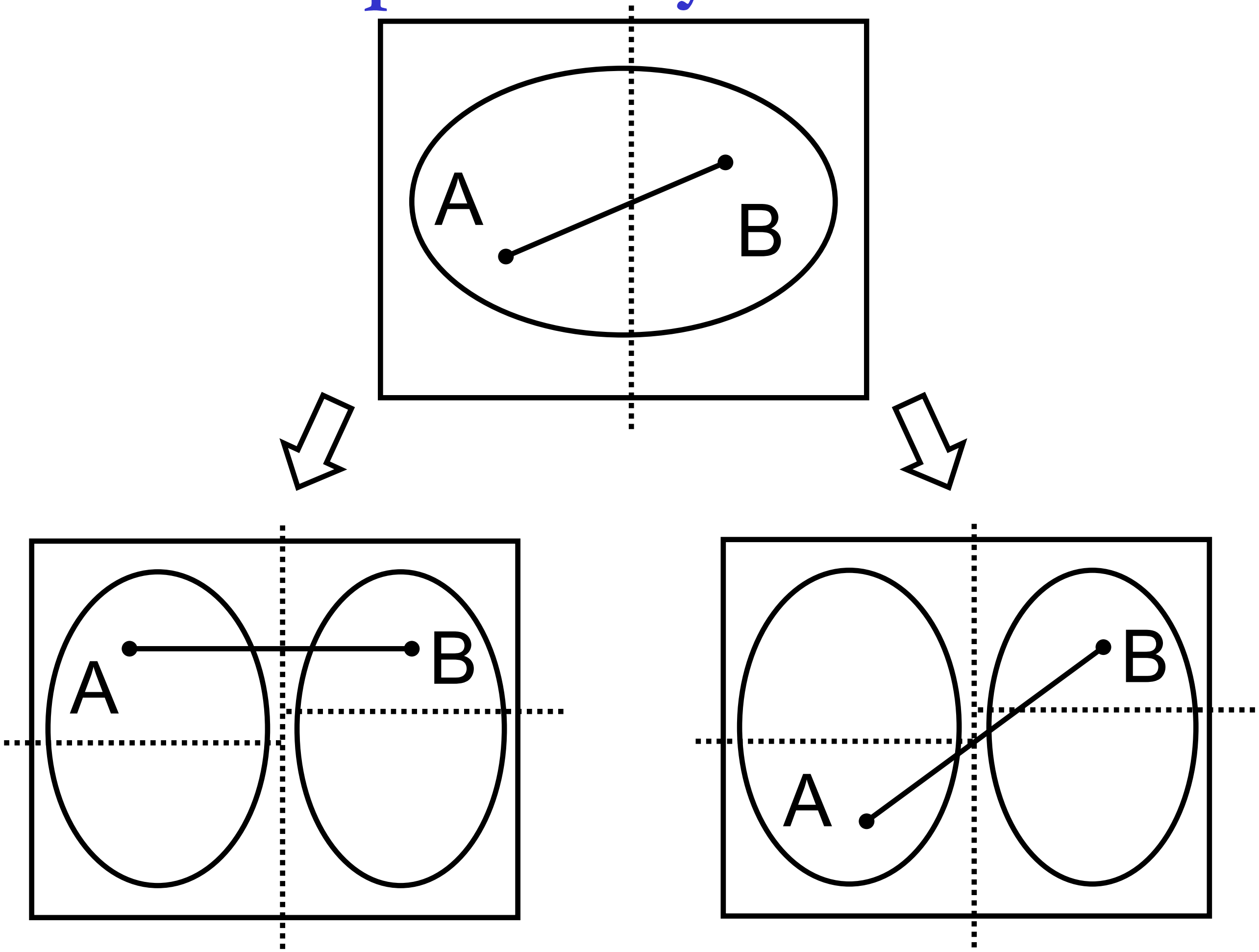






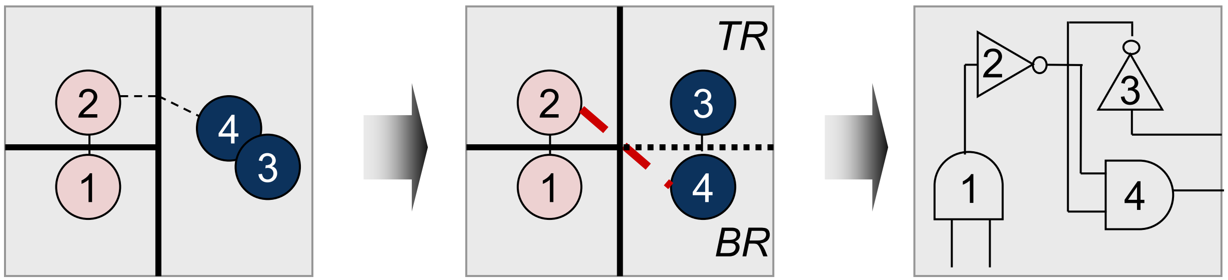


How to Partition Subcircuits "Independently"?

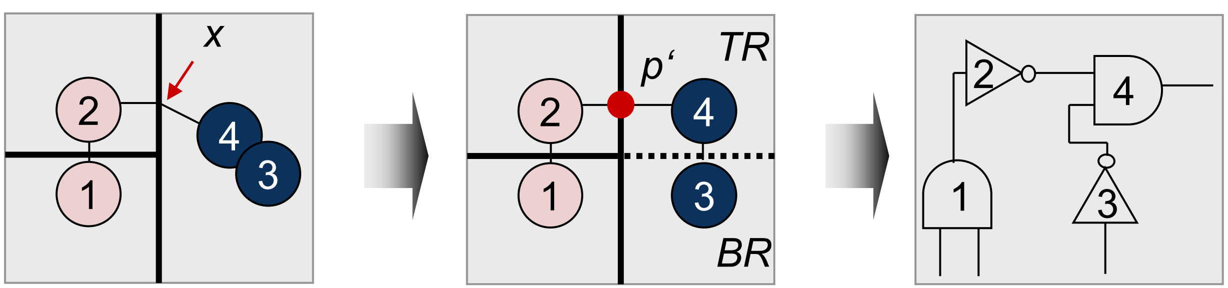


The costs of these two solutions are not the same

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

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

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

- 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

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

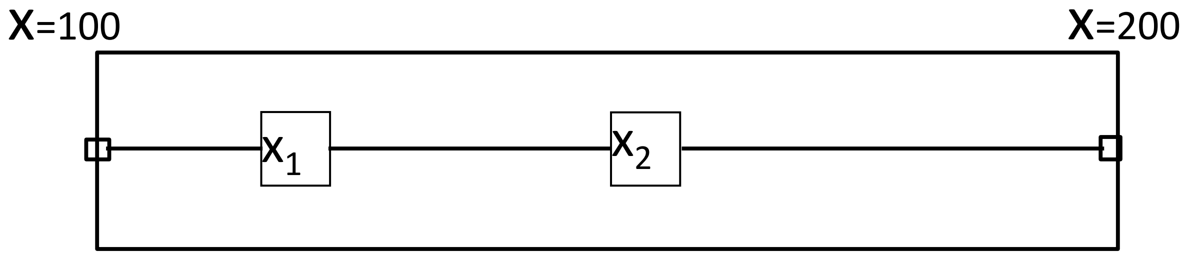
where n is the total number of cells, and $c(i,j)$ is the connection cost between cells i and j.

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

$$\frac{\partial L_x(P)}{\partial X} = AX - b_x = 0 \qquad \frac{\partial L_y(P)}{\partial Y} = AY - b_y = 0$$

- 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 :

$$Ax = b$$

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

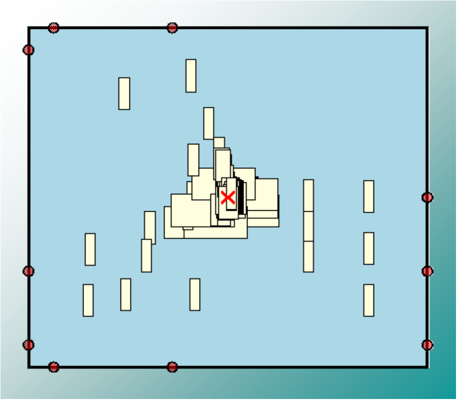
$$\begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 100 \\ 200 \end{bmatrix}$$

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

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

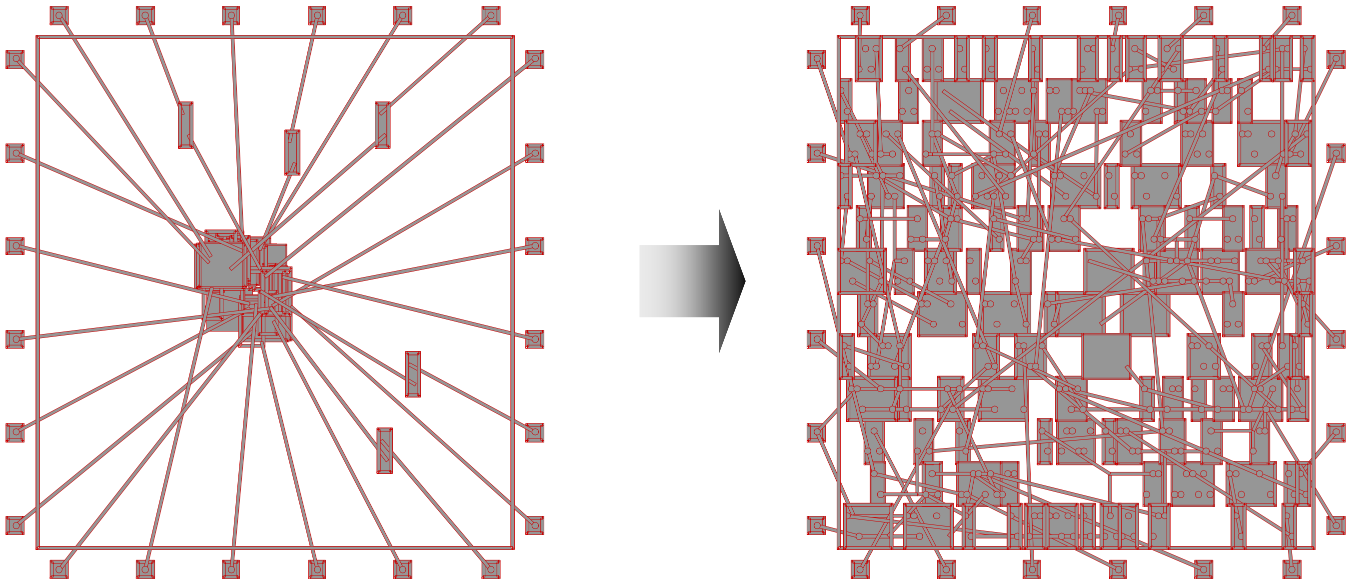
Why "Squared Wirelength" ?

- 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.



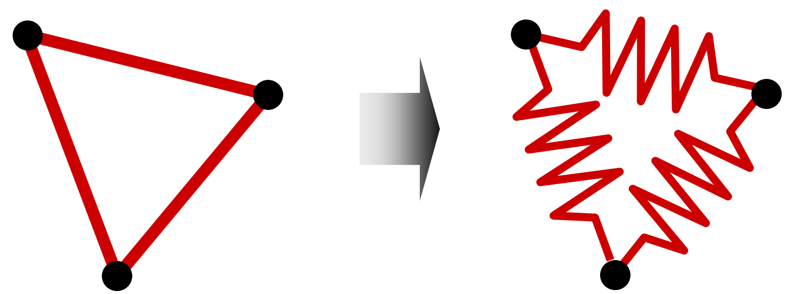
Analytic Placement – Quadratic Placement

- 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.

5 min break

Analytic Placement – Force-directed Placement

- Cells and wires are modeled using the mechanical analogy of a mass-spring system, i.e., masses connected to Hooke's-Law springs



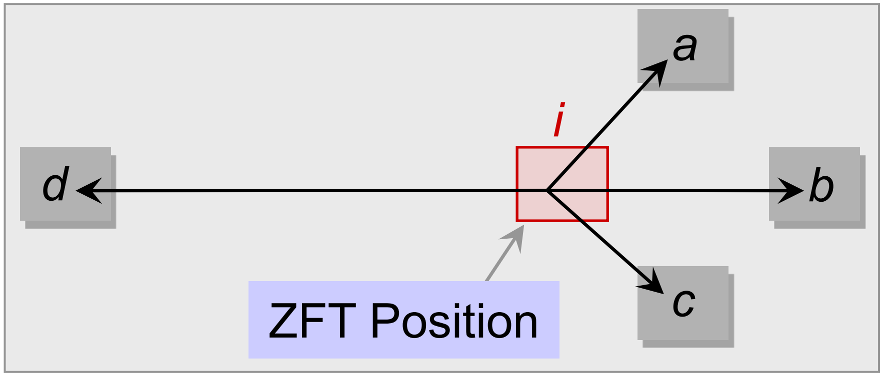
- Attraction force between cells is directly proportional to their distance
- Cells will eventually settle in a force equilibrium → minimized wirelength

Analytic Placement – Force-directed Placement

- Given two connected cells *a* and *b*, the attraction force exerted on *a* by *b* is $\overrightarrow{F_{ab}} = c(a,b) \cdot (\vec{b} - \vec{a})$
  - *c*(*a*,*b*) is the connection weight (priority) between cells *a* and *b*, and
  - $(\vec{b} - \vec{a})$ is the vector difference of the positions of *a* and *b* in the Euclidean plane
- The sum of forces exerted on a cell *i* connected to other cells 1… *j* is $\overrightarrow{F_i} = \sum_{c(i,j)\neq 0} \overrightarrow{F_{ij}}$
- Zero-force target (ZFT): position that minimizes this sum of forces

Analytic Placement – Force-directed Placement

Zero-Force-Target (ZFT) position of cell *i*



$$\min \vec{F_i} = c(i,a) \cdot (\vec{a} - \vec{i}\,) + c(i,b) \cdot (\vec{b} - \vec{i}\,) + c(i,c) \cdot (\vec{c} - \vec{i}\,) + c(i,d) \cdot (\vec{d} - \vec{i}\,)$$

Analytic Placement – Force-directed Placement

Basic force-directed placement

- Iteratively moves all cells to their respective ZFT positions
- *x*- and *y*-direction forces are set to zero:

$$\sum_{c(i,j)\neq 0} c(i,j)\cdot(x_j^0 - x_i^0) = 0 \qquad \sum_{c(i,j)\neq 0} c(i,j)\cdot(y_j^0 - y_i^0) = 0$$

- Rearranging the variables to solve for x_i^0 and y_i^0 yields

$$x_i^0 = \frac{\sum_{c(i,j)\neq 0} c(i,j)\cdot x_j^0}{\sum_{c(i,j)\neq 0} c(i,j)} \qquad y_i^0 = \frac{\sum_{c(i,j)\neq 0} c(i,j)\cdot y_j^0}{\sum_{c(i,j)\neq 0} c(i,j)}$$

Computation of ZFT position of cell *i* connected with cells 1 … *j*

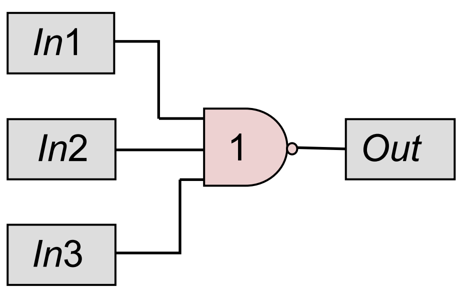
Analytic Placement – Force-directed Placement

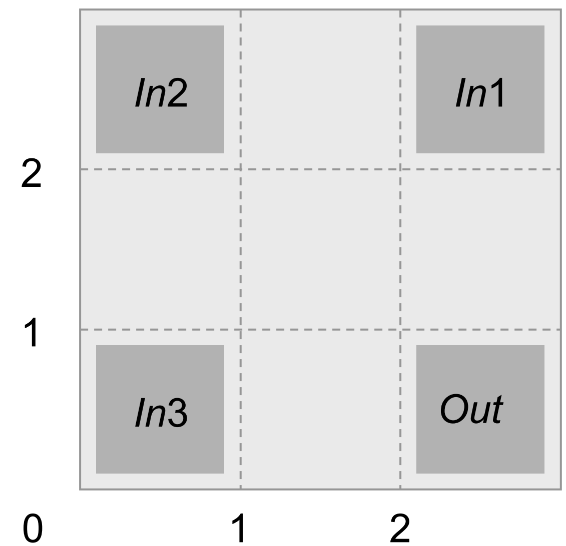
Example: ZFT position

Given:

- Circuit with NAND gate 1 and four I/O pads on a 3 x 3 grid
- Pad positions: *In*1 (2,2), *In*2 (0,2), *In*3 (0,0), *Out* (2,0)
- Weighted connections: $c(a,In1) = 8$, $c(a,In2) = 10$, $c(a,In3) = 2$, $c(a,Out) = 2$

Task: find the ZFT position of cell *a*





Analytic Placement – Force-directed Placement

Example: ZFT position

Given:

- Circuit with NAND gate 1 and four I/O pads on a 3 x 3 grid
- Pad positions: *In*1 (2,2), *In*2 (0,2), *In*3 (0,0), *Out* (2,0)

Solution:

$$x_a^0 = \frac{\sum_{c(i,j)\neq 0} c(a,j)\cdot x_j^0}{\sum_{c(i,j)\neq 0} c(a,j)} = \frac{c(a,In1)\cdot x_{In1} + c(a,In2)\cdot x_{In2} + c(a,In3)\cdot x_{In3} + c(a,Out)\cdot x_{Out}}{c(a,In1)+c(a,In2)+c(a,In3)+c(a,Out)} = \frac{8\cdot 2+10\cdot 0+2\cdot 0+2\cdot 2}{8+10+2+2} = \frac{20}{22} \approx 0.9$$

$$y_a^0 = \frac{\sum_{c(i,j)\neq 0} c(a,j)\cdot y_j^0}{\sum_{c(i,j)\neq 0} c(a,j)} = \frac{c(a,In1)\cdot y_{In1} + c(a,In2)\cdot y_{In2} + c(a,In3)\cdot y_{In3} + c(a,Out)\cdot y_{Out}}{c(a,In1)+c(a,In2)+c(a,In3)+c(a,Out)} = \frac{8\cdot 2+10\cdot 2+2\cdot 0+2\cdot 0}{8+10+2+2} = \frac{36}{22} \approx 1.6$$

ZFT position of cell *a* is (1,2)

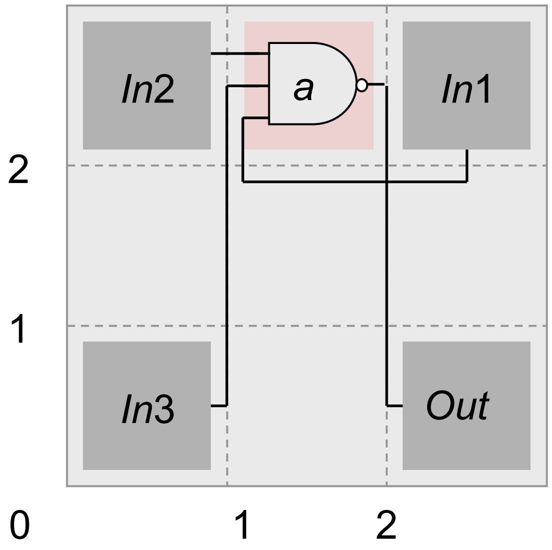
Analytic Placement – Force-directed Placement

Example: ZFT position

Given:

- Circuit with NAND gate 1 and four I/O pads on a 3 x 3 grid
- Pad positions: *In*1 (2,2), *In*2 (0,2), *In*3 (0,0), *Out* (2,0)

Solution:



ZFT position of cell *a* is (1,2)

Analytic Placement – Force-directed Placement

Input: set of all cells *V*
Output: placement *P*

```
P = PLACE(V)                          // arbitrary initial placement
loc = LOCATIONS(P)                    // set coordinates for each cell in P
foreach (cell c ∈ V)
   status[c] = UNMOVED
while (ALL_MOVED(V) || !STOP())       // continue until all cells have been
                                      //   moved or some stopping
                                      //   criterion is reached
    c = MAX_DEGREE(V,status)          // unmoved cell that has largest
                                      //   number of connections
   ZFT_pos = ZFT_POSITION(c)          // ZFT position of c
   if (loc[ZFT_pos] == Ø)             // if position is unoccupied,
      loc[ZFT_pos] = c                //   move c to its ZFT position
   else
      RELOCATE(c,loc)                 //
   status[c] = MOVED                  // mark c as moved
```

Analytic Placement – Force-directed Placement

- Advantages:
  - Conceptually simple, easy to implement
  - Primarily intended for global placement, but can also be adapted to detailed placement
- Disadvantages:
  - Does not scale to large placement instances
  - Is not very effective in spreading cells in densest regions
  - Poor trade-off between solution quality and runtime
- In practice, FDP is extended by specialized techniques for cell spreading
  - This facilitates scalability and makes FDP competitive
  - E.g., add repulsive forces between unconnected cells to reduce overlaps

Modern Placement Algorithms

- Predominantly analytic algorithms
- Solve two challenges: interconnect minimization and cell overlap removal (spreading)
- Two families:

Quadratic placers

Non-convex optimization placers

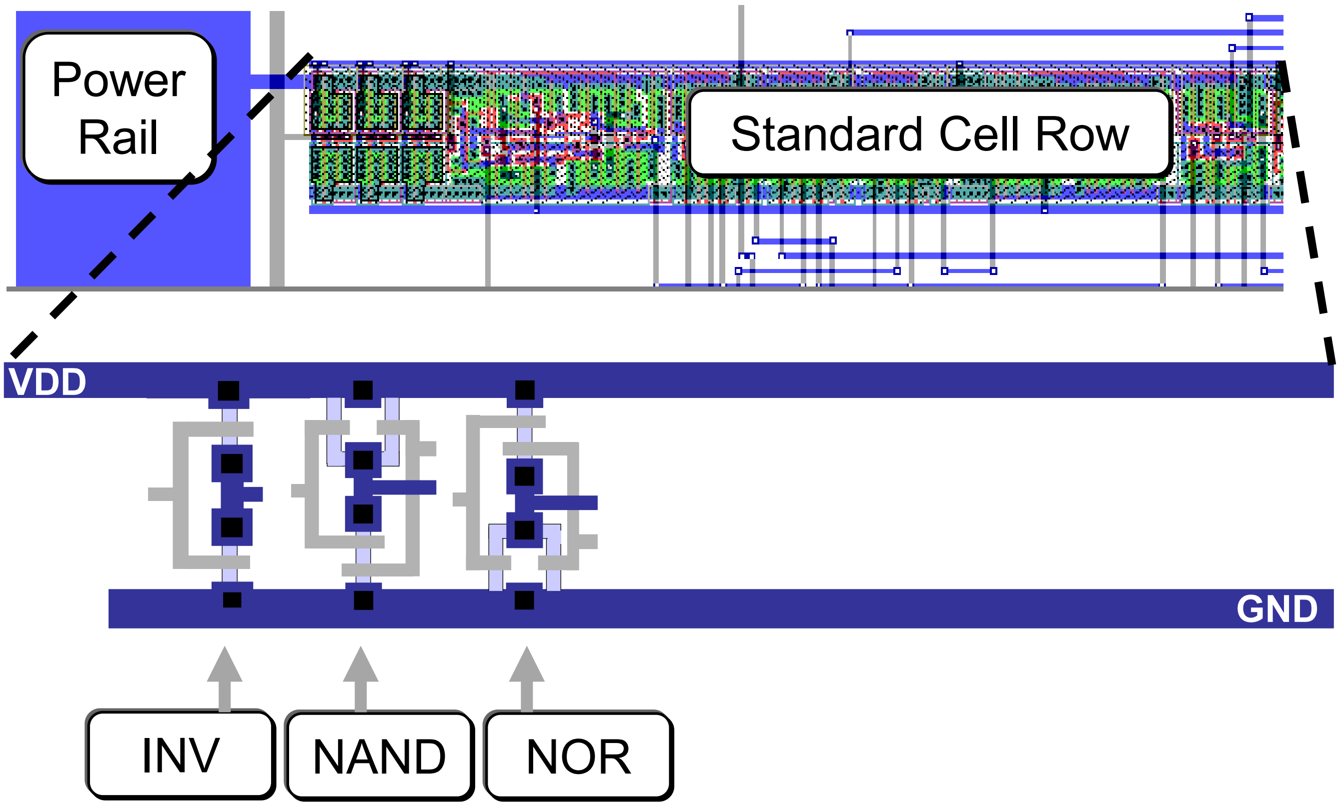
- Quadratic placers are faster, easier to parallelize but non-convex placement usually gives better results

Legalization and Detailed Placement

- Global placement must be legalized
  - Cell locations typically do not align with power rails
  - Small cell overlaps due to incremental changes, such as cell resizing or buffer insertion
- Legalization seeks to find legal, non-overlapping placements for all placeable modules
- Legalization can be improved by detailed placement techniques, such as
  - Swapping neighboring cells to reduce wirelength
  - Sliding cells to unused space
- Software implementations of legalization and detailed placement are often bundled

Legalization and Detailed Placement

Legal positions of standard cells between VDD and GND rails



Summary of Placement– Problem Formulation and Objectives

- Row-based standard-cell placement
  - Cell heights are typically fixed, to fit in rows (but some cells may have double and quadruple heights)
  - Legal cell sites facilitate the alignment of routing tracks, connection to power and ground rails
- Wirelength as a key metric of interconnect
  - Bounding box half-perimeter (HPWL)
  - Cliques and stars
  - RMSTs and RSMTs
- Objectives: wirelength, routing congestion, circuit delay
  - Algorithm development is usually driven by wirelength
  - The basic framework is implemented, evaluated and made competitive on standard benchmarks
  - Additional objectives are added to an operational framework

Summary of Placement– Global Placement

- Combinatorial optimization techniques: min-cut and simulated annealing
  - Can perform both global and detailed placement
  - Reasonably good at small to medium scales
  - SA is very slow, but can handle a greater variety of constraints
  - Randomized and chaotic algorithms – small changes at the input can lead to large changes at the output
- Analytic techniques: QP, force-directed placement and non-convex optimization
  - Primarily used for global placement
  - Unrivaled for large netlists in speed and solution quality
  - Capture the placement problem by mathematical optimization
  - Use efficient numerical analysis algorithms
  - Ensure stability: small changes at the input can cause only small changes at the output
  - Example: a modern, competitive analytic global placer takes 20mins for global placement of a netlist with 2.1M cells (single thread, 3.2GHz Intel CPU)

Summary of Legalization and Detailed Placement

- Legalization ensures that design rules & constraints are satisfied
  - All cells are in rows
  - Cells align with routing tracks
  - Cells connect to power & ground rails
- Detailed placement reduces interconnect, while preserving legality
  - Swapping neighboring cells, rotating groups of three
  - Optimal branch-and-bound on small groups of cells
  - Sliding cells along their rows
- Extensions to optimize routed wirelength, routing congestion and circuit timing
- Relatively straightforward algorithms, but high-quality, fast implementation is important
- Most relevant after analytic global placement, but are also used after min-cut placement
- Rule of thumb: 50% runtime is spent in global placement, 50% in detailed placement