

物理设计

Placement (布局)

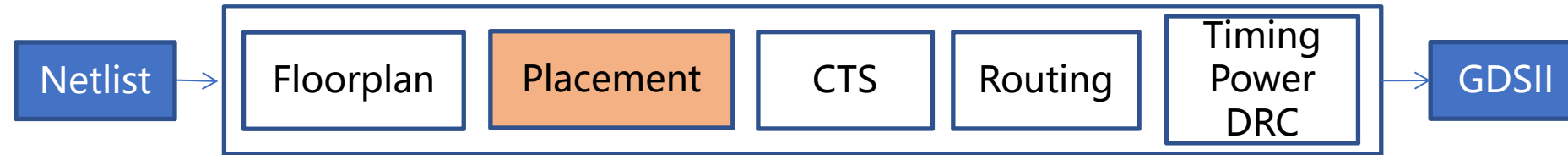
李兴权

iEDA

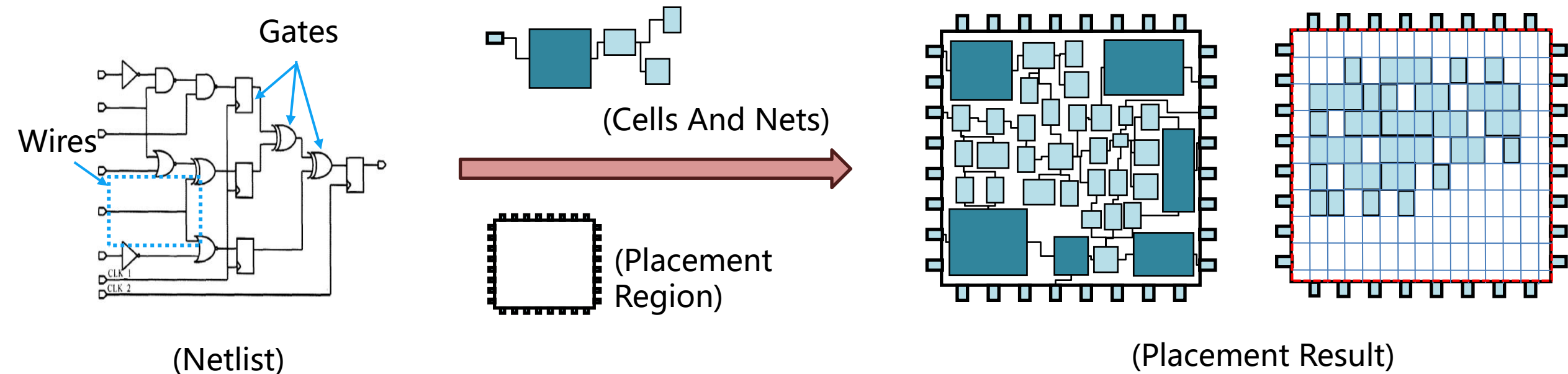
- 01** **Introduction**
- 02** **Placement Techniques**
- 03** **Global Placement**
- 04** **Detail Placement**
- 05** **Timing Driven Placement**
- 06** **Congestion Driven Placement**

Placement

- Placement in physical design

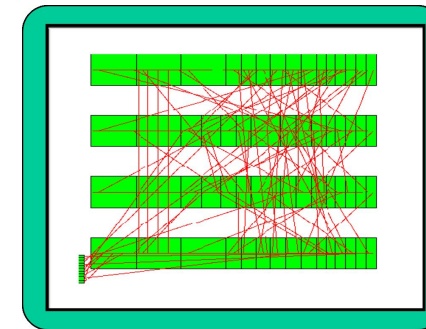
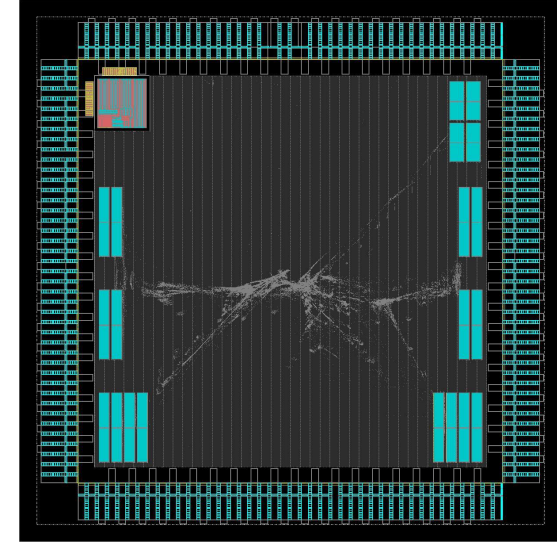


- What placement means?

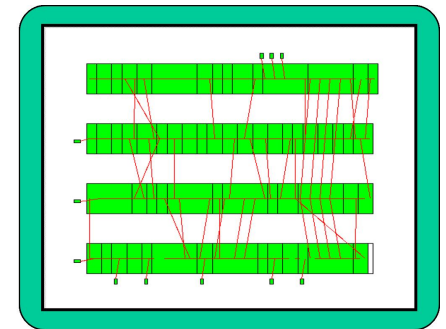


Placement Task

- Design Planning is completed
 - Core areas defined
 - Macros are placed and “fixed”
 - Placement blockages defined
 - Power grid pre-routed
- Task
 - Decide the good location of standard cell,
 - Satisfy some constraints.
- Objectives
 - Wirelength, 线长
 - Timing, 时序
 - Routability, 可布线行等
- Constraints
 - Core region, Cell density, Pin density, Fence region, Row/site alignment, ...



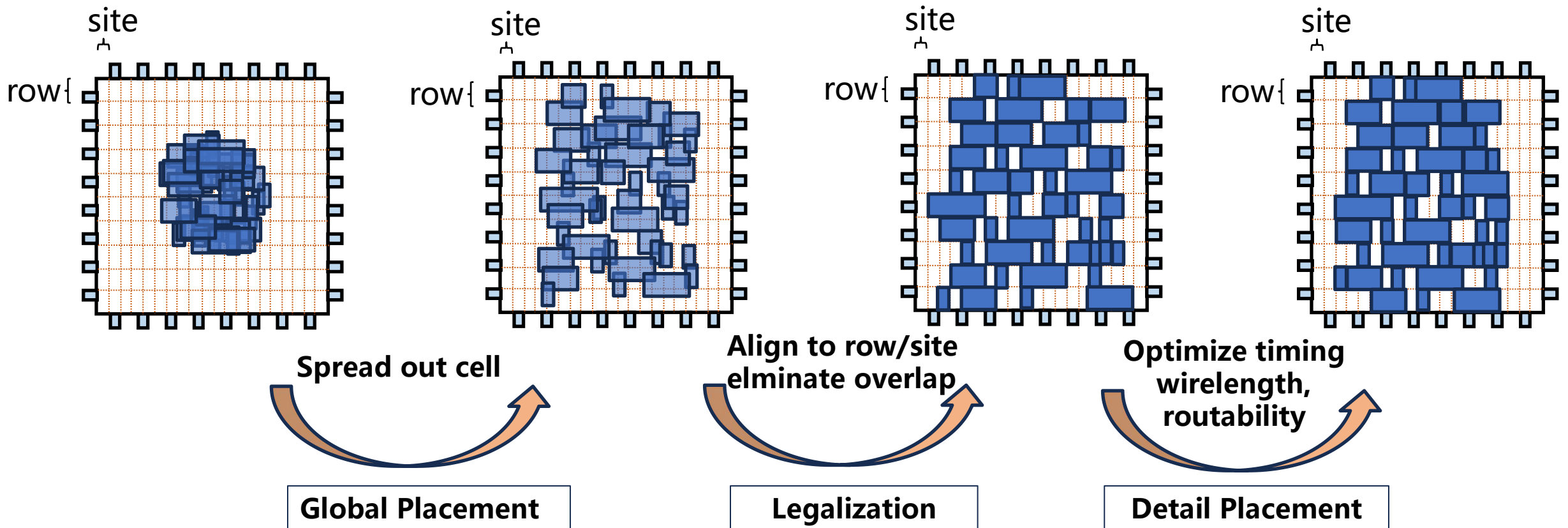
Bad Placement



Good Placement

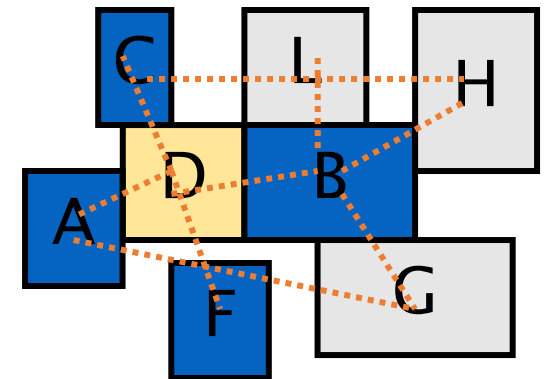
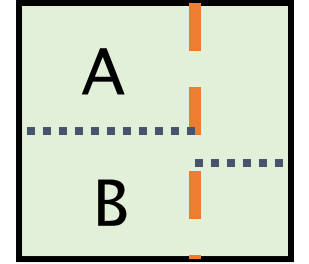
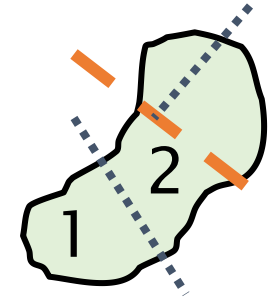
Placement Steps

- Placement needs to promise that cells do not overlap and are assigned to row/site

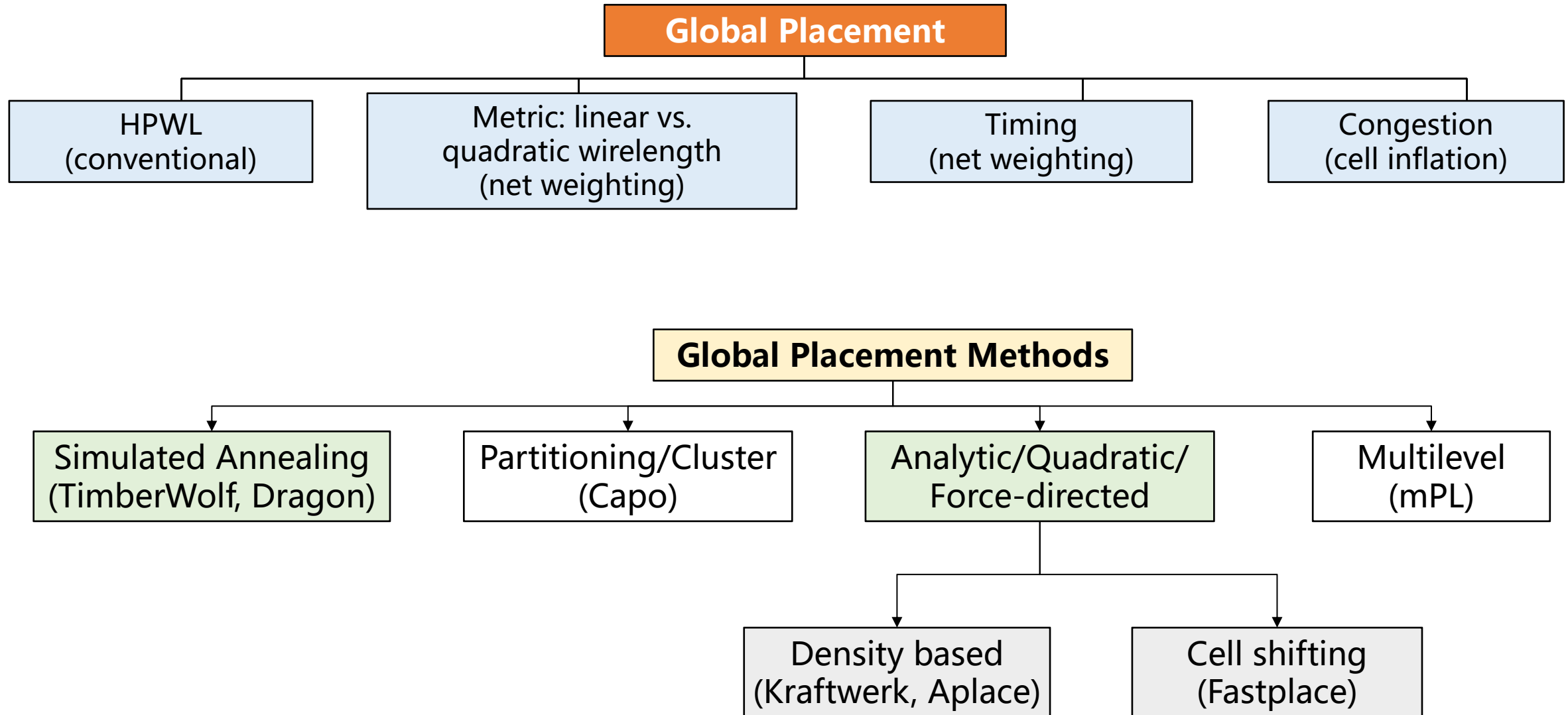


Placement Methods

- - 基于划分的布局
 - - 递归的二分划分或四分划分
 - - 切割方向?
 - - 划分与物理位置
- - 迭代法
 - - 模拟退火法 (Simulated Annealing)
 - - 或：力导向/解析法 (Force Directed/Analytic)
 - - 从初始布局开始，迭代地改进布线长度/面积
- - 构造法
 - - 从中心位置放置少量单元，然后将高度连接的相邻模块放置在它们周围



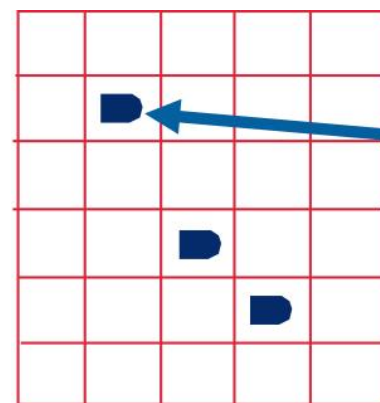
Placement Overview



- 01** Introduction
- 02** Placement Techniques
- 03** Global Placement
- 04** Detail Placement
- 05** Timing Driven Placement
- 06** Congestion Driven Placement

简单布局

- 假设一个非常简单的芯片模型：
 - - 简单网格：单元格按方格排列。
 - - 引脚固定在边缘。
- 假设一个简单的门模型：
 - - 所有门的大小相同。
 - - 每个网格位置只能放置一个门。

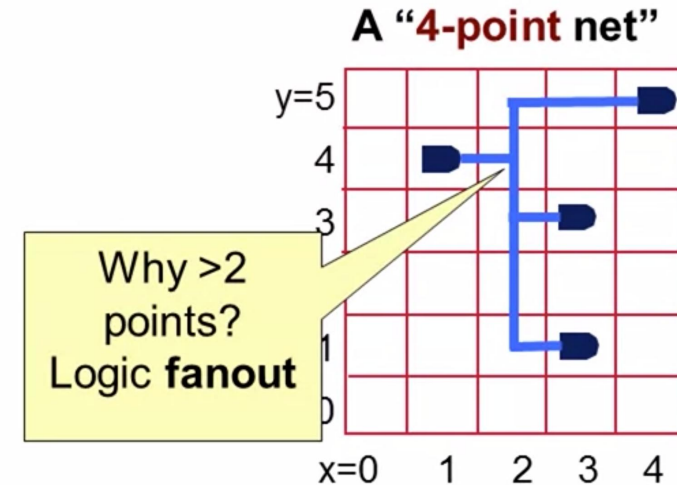
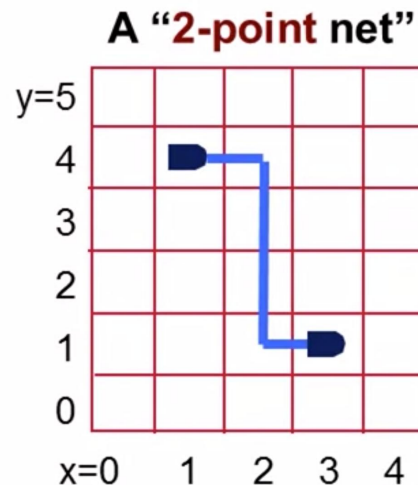


Any gate
can go in
any slot

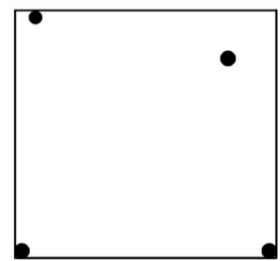


Each cell
can go
anywhere
in any row

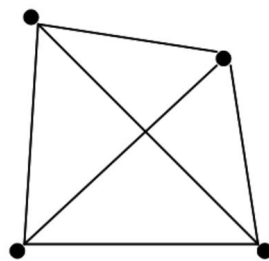
- 假设一个非常简单的芯片模型：
 - - 简单网格：单元格按方格排列。
 - - 引脚固定在边缘。
- 假设一个简单的门模型：
 - - 所有门的大小相同。
 - - 每个网格位置只能放置一个门。



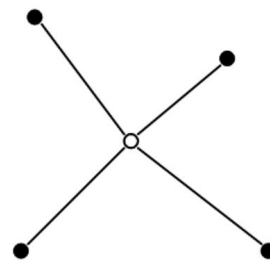
- 简单的边界框布线长度估算器 (Simple Bounding Box Wirelength Estimator) :



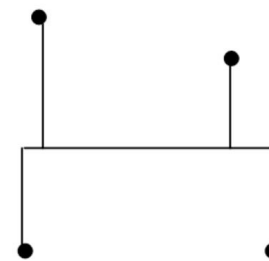
Bounding box model



Clique model



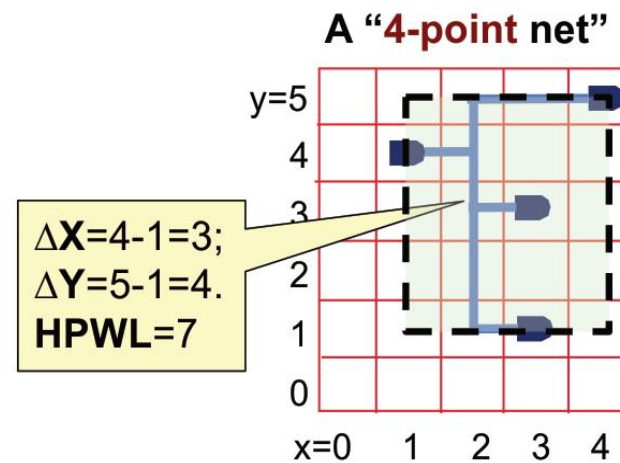
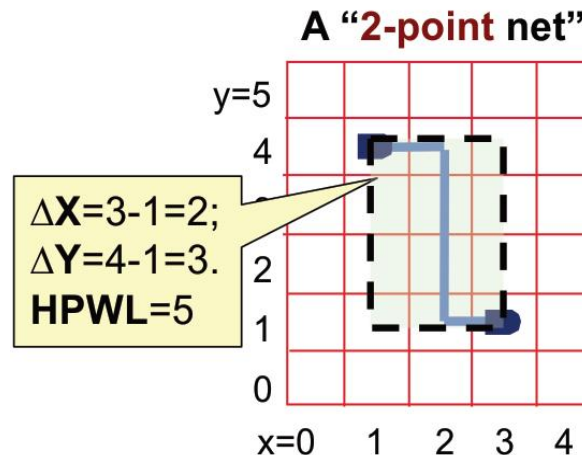
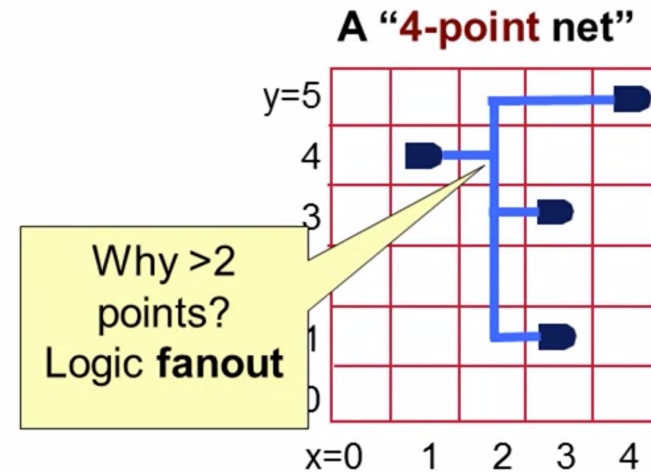
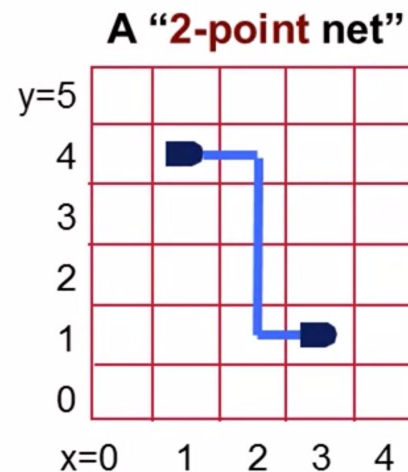
Star model



Steiner model

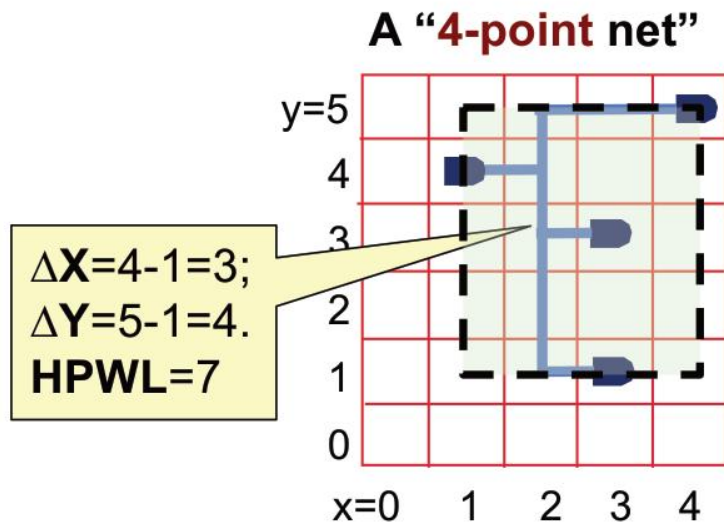
线长评估

- 假设一个非常简单的芯片模型：
 - - 简单网格：单元格按方格排列。
 - - 引脚固定在边缘。
- 假设一个简单的门模型：
 - - 所有门的大小相同。
 - - 每个网格位置只能放置一个门。
- 简单的边界框布线长度估算器 (Simple Bounding Box Wirelength Estimator)
 - “Half-Perimeter Wirelength” (HPWL)
 - Put the smallest box around all gates on net.
 - Measure Width (ΔX) and Height (ΔY) of box.
 - $HPWL = \Delta X + \Delta Y$



半周长线长 (HPWL)

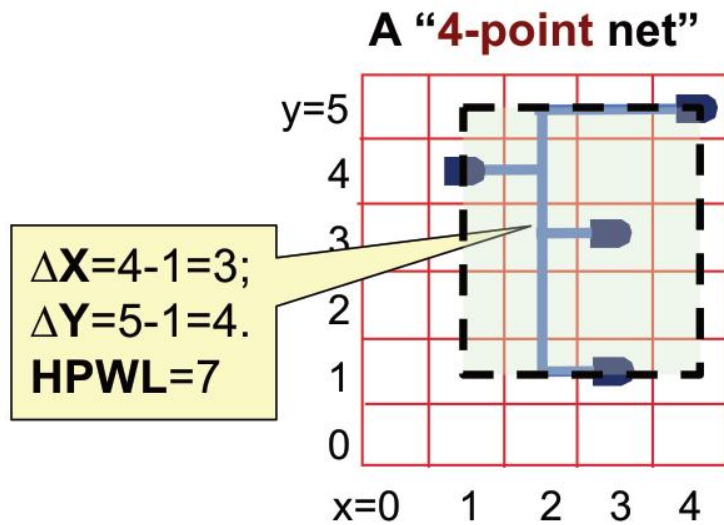
- 半周长线长公式
 - HPWL = 7
 - 真实线长 = ?



$$HPWL(e) = \max_{i,j \in e} |x_i - x_j| + \max_{i,j \in e} |y_i - y_j|$$

半周长线长 (HPWL)

- 半周长线长公式
 - HPWL = 7
 - 真实线长 = 9



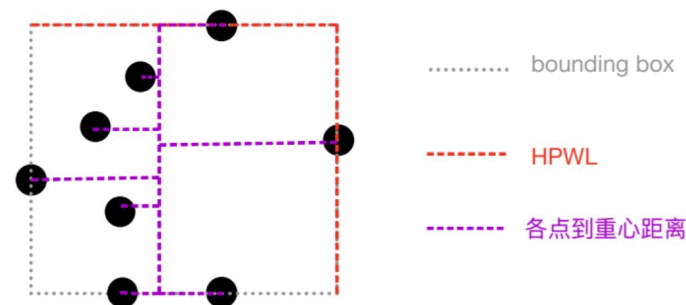
$$HPWL(e) = \max_{i,j \in e} |x_i - x_j| + \max_{i,j \in e} |y_i - y_j|$$

■ P2WWL

$$New\ wl = \max_{i,j \in e} |y_i - y_j| + \sum_{i \in e} |x_i - gravity|$$

优点:

- 能比HPWL更准确描述多pin点线网线长



存在问题:

- 重心线选择在x/y方向可能造成较大误差
- 计算比HPWL耗时 (思考: 能否在last 20%阶段替换HPWL)
- 未能合适的对线长函数“光滑化”处理

LP范数作光滑
前面一项用LSE/WA 后面一项用LP

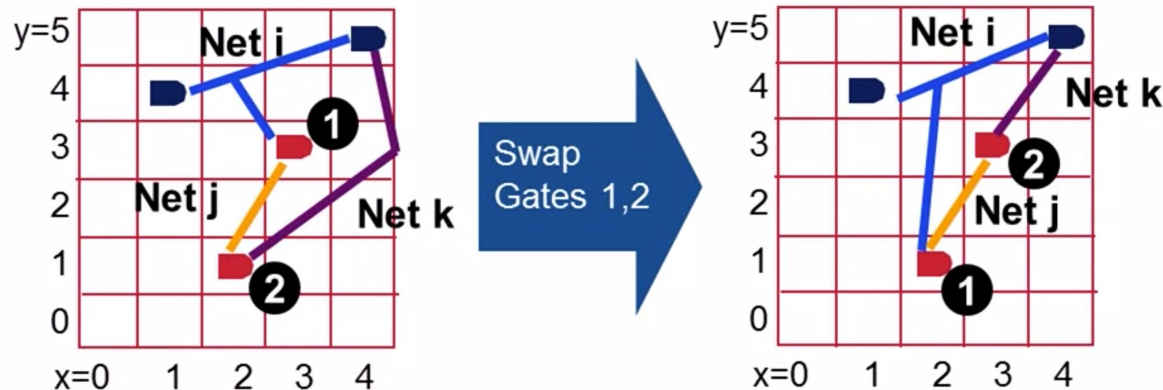
半周长线长 (HPWL)

$$HPWL(e) = \max_{i,j \in e} |x_i - x_j| + \max_{i,j \in e} |y_i - y_j|$$

- Easy to calculate, even for a multi-point net
- Always a lower bound on the real wire length
 - No matter how complex the final routed wire path is
 - you need at least this much wire to connect everything.
 - Aside: all wiring, on big chips (and most wire on boards) is strictly horizontal & vertical - no "arbitrary angles" for manufacturing reasons.
- Another reason HPWL is a good estimator.

迭代布局

- Start with a random placement:
 - Randomly assign each gate to a grid location.
- Apply random iterative improvement:
 - Pick a random pair of gates.
 - Swap their locations and evaluate the change in wirelength.
 - If wirelength got smaller – accept the swap.
 - If wirelength got bigger – undo the swap.
 - Repeat until wirelength stops improving.



```
// Random initial placement
foreach (gate Gi in netlist)
    place Gi in random (x,y) not occupied.

// calculate initial HPWL
L=0
foreach (net Ni in netlist)
    L = L + HPWL(Ni)

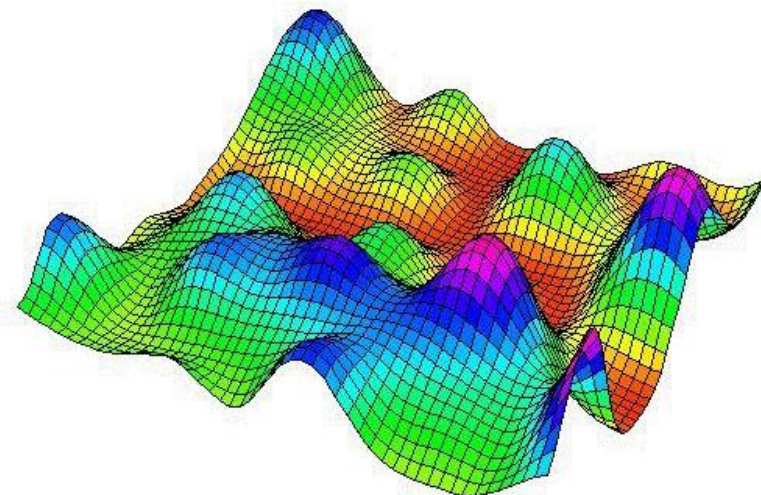
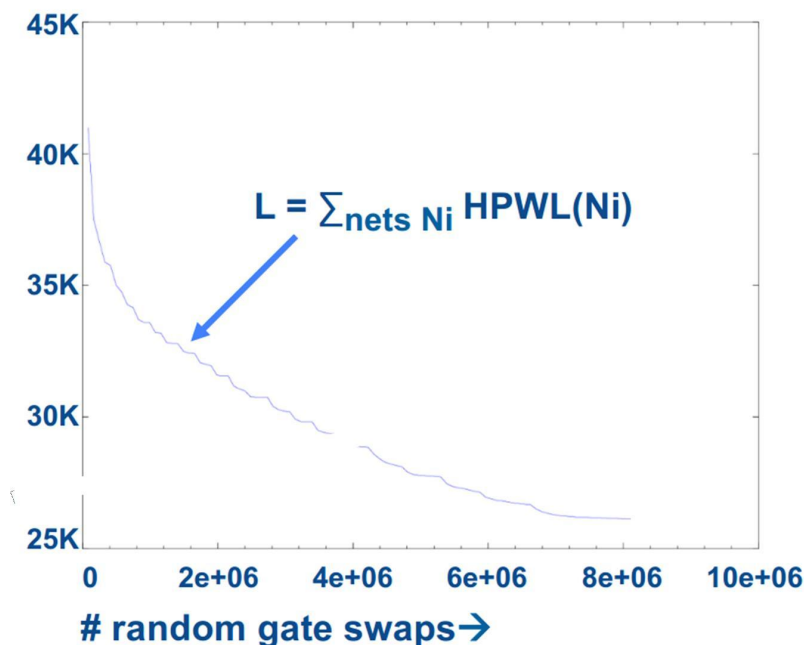
// random iterative improvement
while (overall HPWL is improving)
    pick two random gates Gi, Gj
    swap Gi and Gj
    evaluate  $\Delta L = \text{new HPWL} - \text{old HPWL}$ 
    if ( $\Delta L < 0$ ) then keep the swap
    if ( $\Delta L > 0$ ) undo the swap
```

- It works ok....

- Small benchmark
- ~2500 gates + ~500 pins
- Placed in a 50x50 grid

- Graph shows “progress”

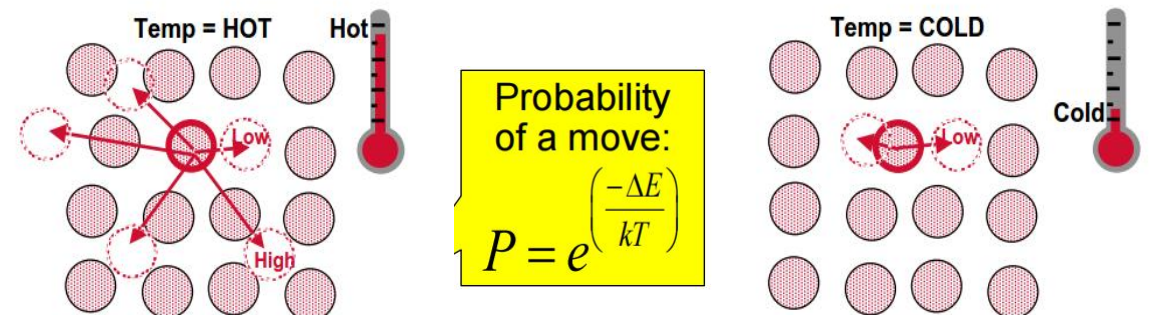
- Y axis: $L = \sum_{\text{nets } Ni} \text{HPWL}(Ni)$
- X axis: # swaps
- Yes, this ran for **8M swaps** until progress on L stopped



模拟退火 (SA)

- Start with the same basic algorithm:
 - Random initial placement
 - Swap two Random gates
 - Evaluate change in HPWL (ΔL)
 - If wirelength improves, accept the change.
- But what if wirelength increases ($\Delta L > 0$)?
 - Evaluate annealing probability function
 - Choose a uniform random number (R)
 - between 0 and 1
 - If $R < P$ then keep the swap.
- What is T?
 - T is the simulated temperature.
 - Start hot. Cool down

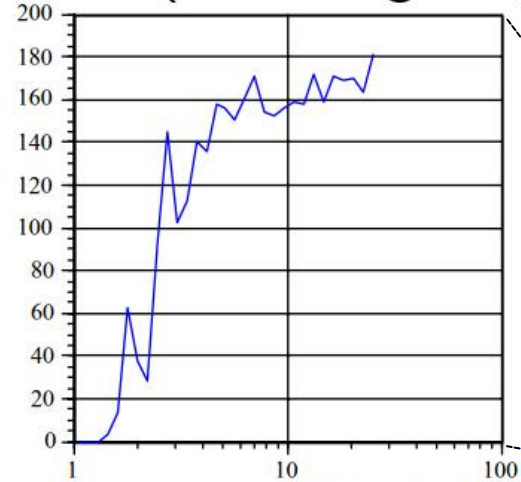
```
T=HOT; frozen = false
while (!frozen)
    swap two random gates Gi, Gj
    evaluate  $\Delta L$ 
    if ( $\Delta L < 0$ ) then
        keep the swap
    else
        if (random() <  $\exp(-\Delta L/T)$ )
            accept swap
        else
            undo swap
    if (HPWL still decreasing)
        T = 0.9*T
    else
        frozen=true
```



模拟退火 (SA)

- How well does this work?
 - Really well!
 - Many EDA algorithms
 - use Simulated Annealing.
- Does it find an optimal solution?
 - No. But it's good at avoiding local minima.
- What happens if I run it again?
 - I will get a different answer each time!
- NOT how placers work today

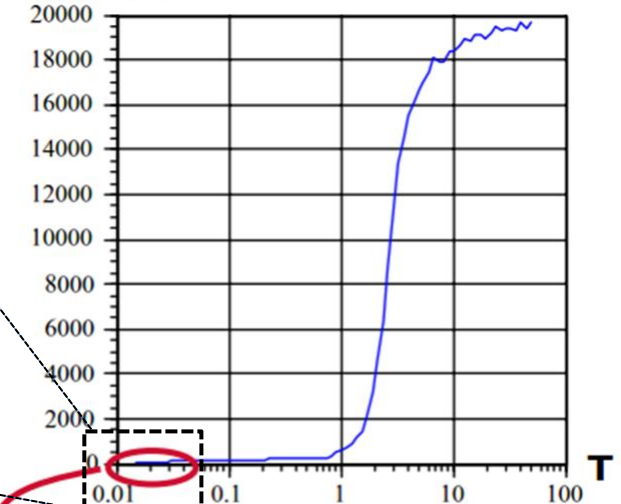
Final cost (end of moves @ each T)



Temperature
on log scale

← annealing

Final cost



Final cost != 0, = 61

Analytical Placement (分析方法)

- Question:
 - Can we write an equation whose minimum is the placement?
 - If we have a cost function (such as wirelength) that is a function of the gate coordinates (x, y) ,

$$WL = f(x, y)$$

where $x = (x_1, x_2, \dots, x_n)^T$, $y = (y_1, y_2, \dots, y_n)^T$

$$\frac{\partial f}{\partial x} = 0, \frac{\partial f}{\partial y} = 0$$

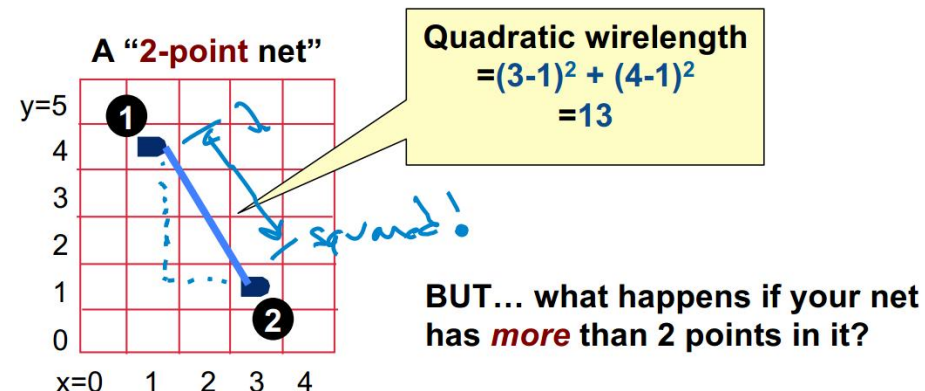
- Then we could find the minimum of f and this would be our optimal placement (x^*, y^*) .
- Sounds crazy, but the answer is YES!
 - All modern placers are based on analytical placement.
 - We need to write the cost function in a mathematically friendly way.
 - Then, we can just differentiate and equate to 0!

Analytical Placement Objective Function (目标函数)

- Instead of HPWL, let's define a new wirelength model

- Quadratic wirelength:
 - 2-point net

$$WL = f(x, y) = (x_1 - x_2)^2 + (y_1 - y_2)^2$$

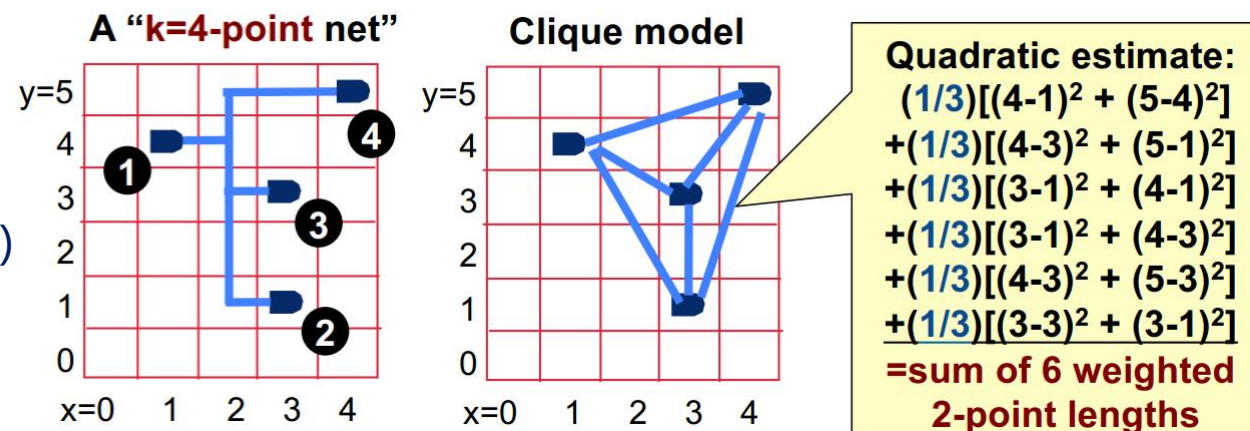


- What about a k-point net ($k > 2$)?

- Instead of one "real" net, replace with a fully connected clique model.
- So each gate on the net has a one-to-one connection with the other.
- Altogether $k(k-1)/2$ nets
- Compensate by weighting each net by $1/(k-1)$

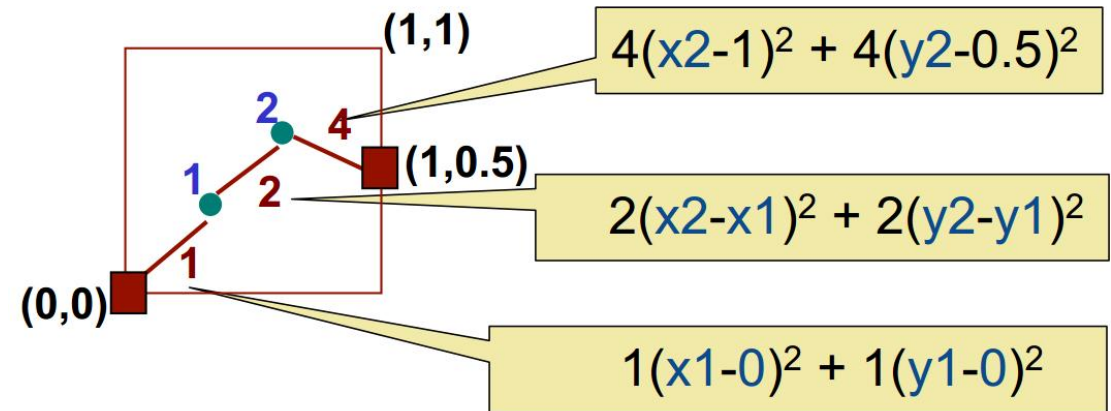
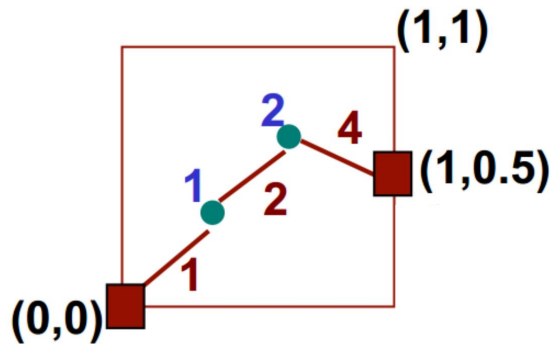
- One last point:

- Assume that gates are dimensionless points



Analytical Placement Calculation

- Example of quadratic wirelength calculation:
 - **2 gate point**, each point has an unknown (x_i, y_i) coordinate, $[0, 1]$.
 - **3 nets**, each net has a pre-assigned weight, (1, 2, 4)
 - **2 pads**, pads are fixed pins on the edge.

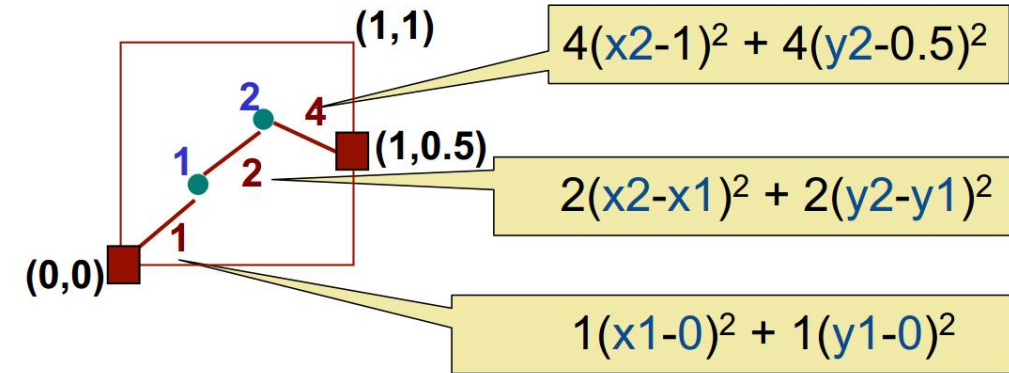


- Important:
 - There are no terms with $x_i \times y_i$
 - Therefore, we can separate x and y terms in the sum

Analytical Placement Calculation

- Now that we have an analytic expression for the cost function, we can use basic calculus to minimize it!

$$\frac{\partial f}{\partial x} = 0, \frac{\partial f}{\partial y} = 0$$



$$Q(x) = 4(x_2 - 1)^2 + 2(x_2 - x_1)^2 + 1(x_1 - 0)^2$$



$$\frac{\partial Q(x)}{\partial x_1} = 0 + 4(x_2 - x_1)(-1) + 2(x_1) = 6x_1 - 4x_2 = 0$$

$$\frac{\partial Q(x)}{\partial x_2} = 8(x_2 - 1) + 4(x_2 - x_1) + 0 = -4x_1 + 12x_2 - 8 = 0$$

$$\begin{pmatrix} 6 & -4 \\ -4 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 8 \end{pmatrix}$$

$$\begin{aligned} x_1 &= 0.571 \\ x_2 &= 0.857 \end{aligned}$$

$$Q(y) = 4(y_2 - 0.5)^2 + 2(y_2 - y_1)^2 + 1(y_1 - 0)^2$$



$$\frac{\partial Q(y)}{\partial y_1} = 0 + 4(y_2 - y_1)(-1) + 2(y_1) = 4y_1 - 4y_2 = 0$$

$$\frac{\partial Q(y)}{\partial y_2} = 8(y_2 - 0.5) + 4(y_2 - y_1) + 0 = -4y_1 + 12y_2 - 4 = 0$$

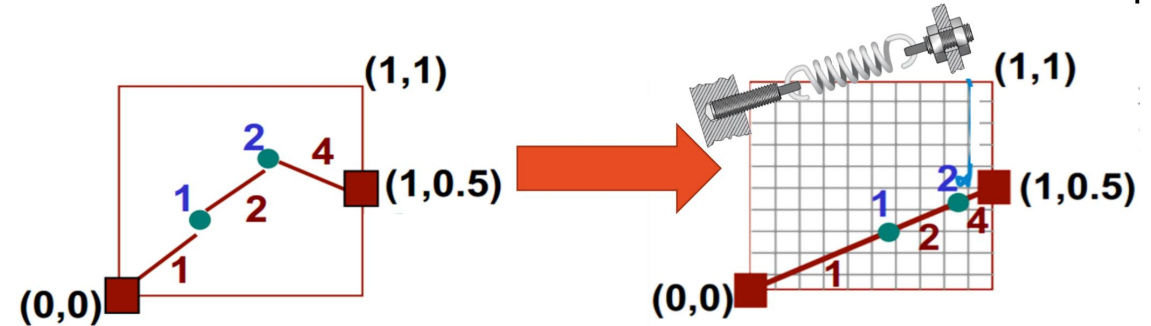
$$\begin{pmatrix} 6 & -4 \\ -4 & 12 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$$

$$\begin{aligned} y_1 &= 0.286 \\ y_2 &= 0.429 \end{aligned}$$

Analytical Placement Calculation

- Observations:

- Placement makes visual sense:
 - All points are on a straight line.
 - Placement minimizes spring weights.
 - Bigger weight $\rightarrow$ Shorter wire



$$\begin{pmatrix} 6 & -4 \\ -4 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 8 \end{pmatrix} \quad \begin{matrix} G_1 : (0.571, 0.286) \\ G_2 : (0.857, 0.429) \end{matrix} \quad \begin{pmatrix} 6 & -4 \\ -4 & 12 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$$

- Algebraically:

- For n gates, we get two equivalent
- $n \times n$ matrices (A) for two linear systems:

$$Ax = b_x, \quad Ay = b_y$$

$$A = \begin{pmatrix} 6 & -4 \\ -4 & 12 \end{pmatrix}$$

$$b_x = \begin{pmatrix} 0 \\ 8 \end{pmatrix}, b_y = \begin{pmatrix} 0 \\ 4 \end{pmatrix}$$

- b_x and b_y vectors represent Pad coordinates!

Building Analytical Network

- Recipe for success

- Build connectivity matrix (C) as follows.

$$C_{i,j} = C_{j,i} = \begin{cases} 0 & i = j \\ 0 & \text{no net}_{i,j} \\ w_{i,j} & \text{net}_{i,j} \end{cases}$$

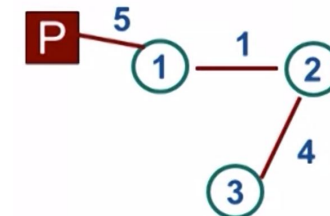
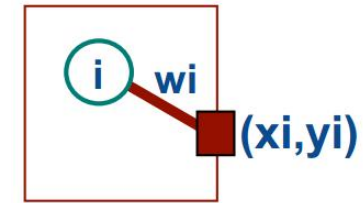
- Build A matrix:

$$A_{i,j} = \begin{cases} -C_{i,j} & i \neq j \\ \sum_{j=1}^N C_{i,j} + w_{j,\text{pads}} & i = j \end{cases}$$

- Build b vectors:

$$b_{x,i} = \begin{cases} 0 & \text{no pad} \\ w_i \cdot x_i & \text{pad } (x_i, y_i, w_i) \end{cases}$$

$$b_{y,i} = \begin{cases} 0 & \text{no pad} \\ w_i \cdot y_i & \text{pad } (x_i, y_i, w_i) \end{cases}$$

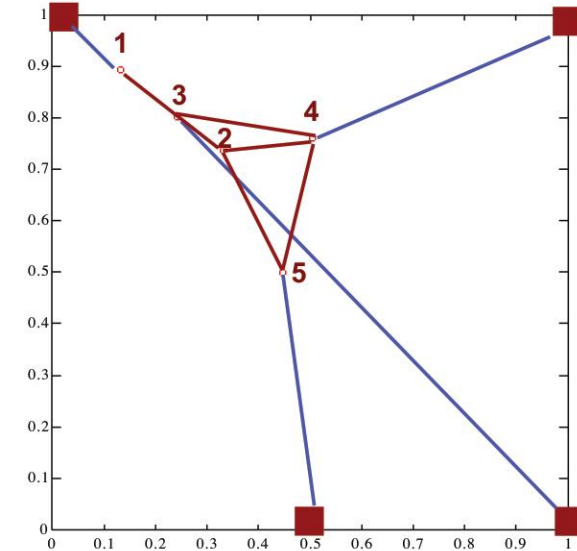
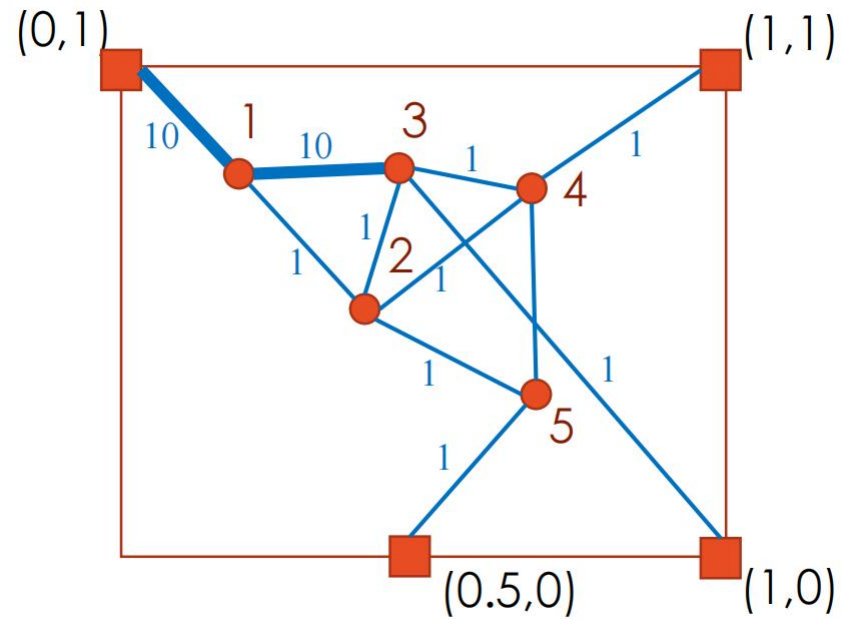


$$C = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 4 \\ 0 & 4 & 0 \end{pmatrix} \quad A = \begin{pmatrix} 6 & -1 & 0 \\ -1 & 5 & -4 \\ 0 & -4 & 4 \end{pmatrix}$$

diagonal

The matrix C has a blue circle around the element 4 in the third row, second column. The matrix A has a red diagonal line drawn through it, and the word 'diagonal' is written to its right. A blue arrow points from the circled 4 in C to the diagonal line in A.

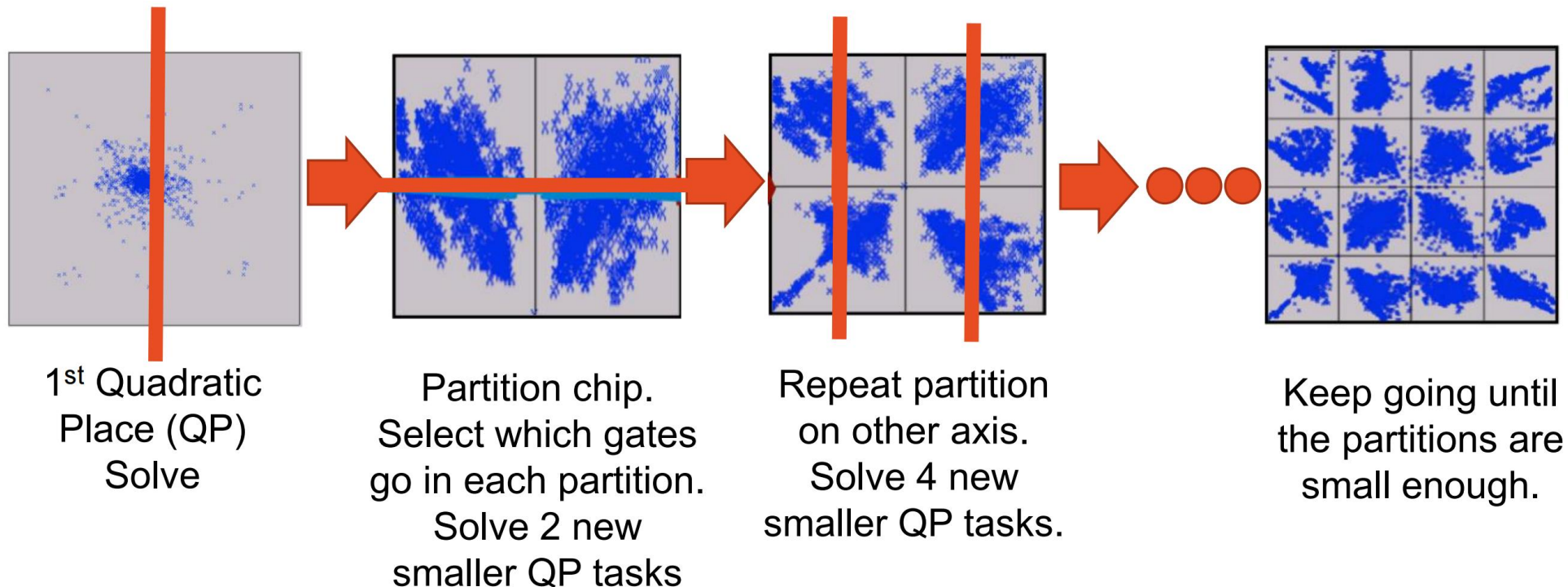
5 Gate Example



$$C = \begin{pmatrix} 0 & 1 & 10 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 10 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \quad A = \begin{pmatrix} 21 & -1 & -10 & 0 & 0 \\ -1 & 4 & -1 & -1 & -1 \\ -10 & -1 & 13 & -1 & 0 \\ 0 & -1 & -1 & 4 & -1 \\ 0 & -1 & 0 & -1 & 3 \end{pmatrix} \quad b_x = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0.5 \end{pmatrix} \quad b_y = \begin{pmatrix} 10 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

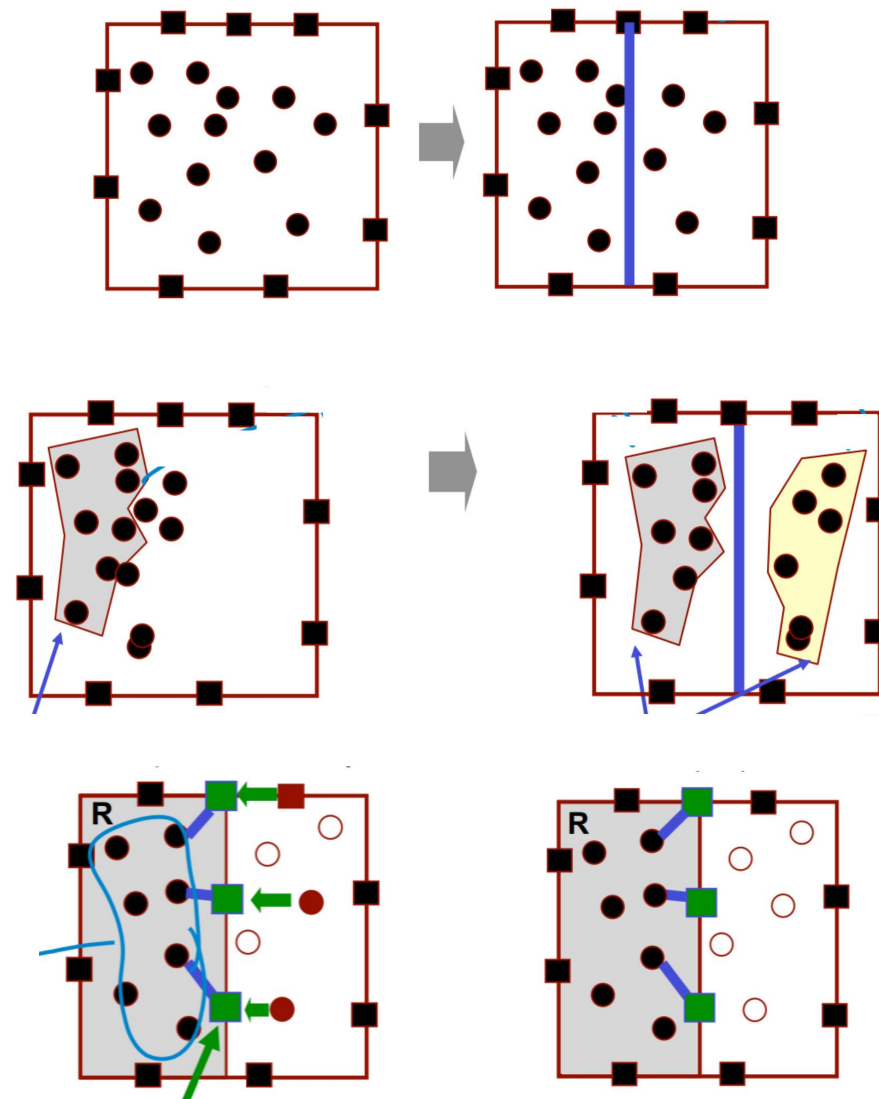
Partition Placement

- What does a real quadratic placement look like?
 - All the gates want to be in the same place!
- How can we solve this?
 - Recursive Partitioning!



Partition Placement (划分布局)

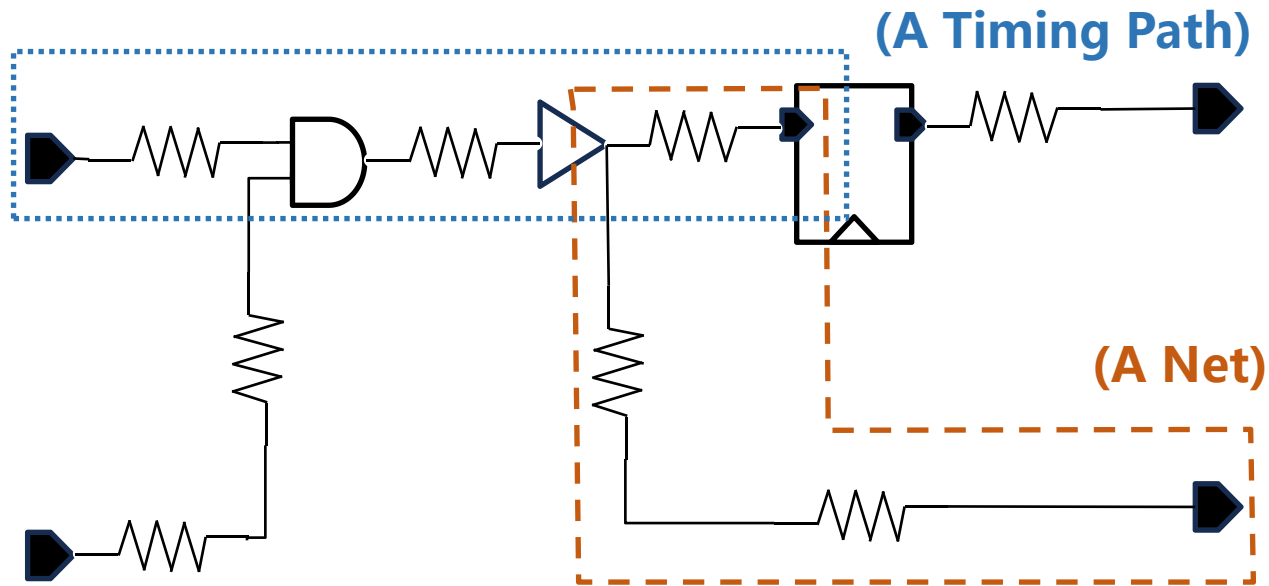
- Partition
 - Divide the chip into new, smaller placement tasks.
 - Divide it in half!
- Assignment
 - Assign gates into new, smaller region.
 - Sort the gates and distribute to each half.
- Containment
 - Formulate new QP matrix that keeps gates in new regions.
 - Create “pseudo pads” –
 - Every gate and pad NOT inside the partition R is modeled as a pad on the boundary of R.
 - Propagate the pseudo pads to the their nearest point on R



- 01** Introduction
- 02** Placement Techniques
- 03** **Global Placement**
- 04** Detail Placement
- 05** Timing Driven Placement
- 06** Congestion Driven Placement

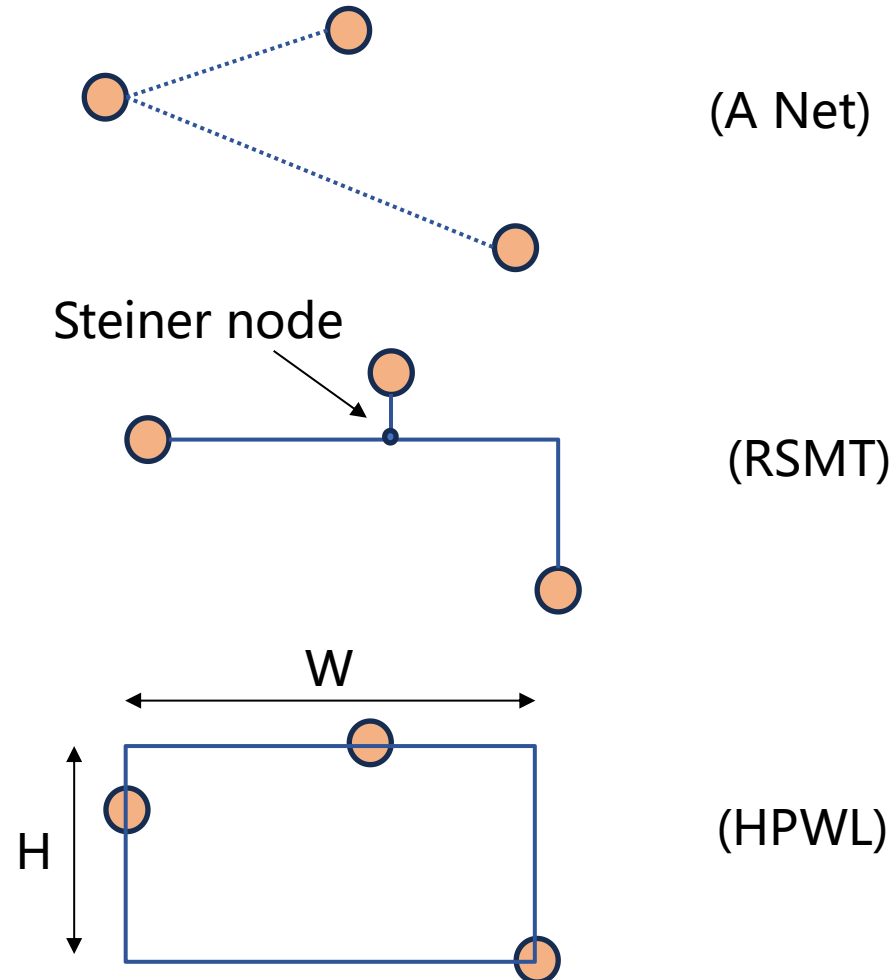
Preliminaries

- Wire connection between cells greatly affects wire length, timing, power...



(Figure to illustrate timing path and net)

- RSMT: Rectilinear Steiner Minimal Tree
- HPWL: Half Perimeter Wirelength

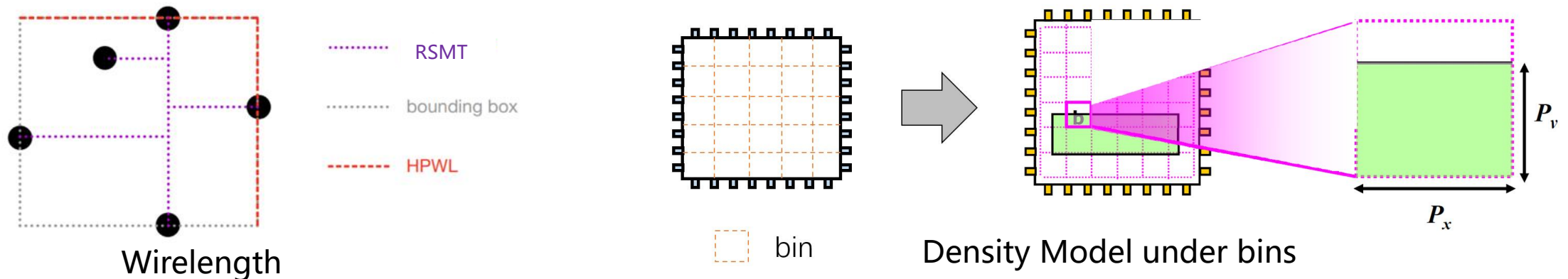


Modern Global Placement

- The global placement is modeled as a constrained optimization problem.
 - Objective is total wirelength. Constraint is the density of each bin

$$\begin{aligned} \min_v \quad & \sum_{e \in E} WL_e(v) \\ \text{s.t.} \quad & \rho_b(v) \leq \rho_0, \forall b \in \text{Bin} \end{aligned}$$

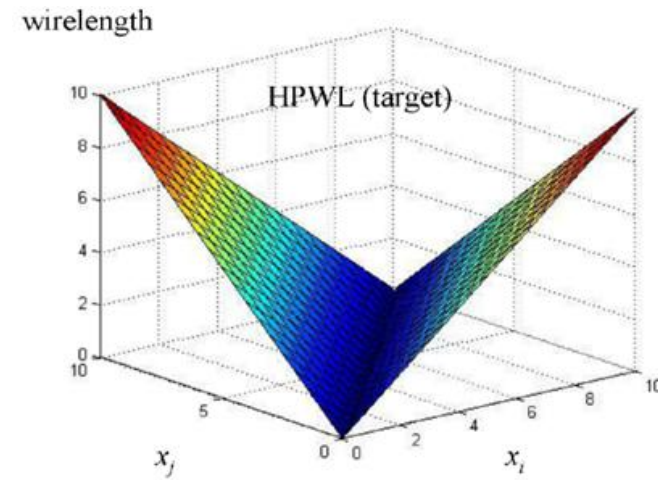
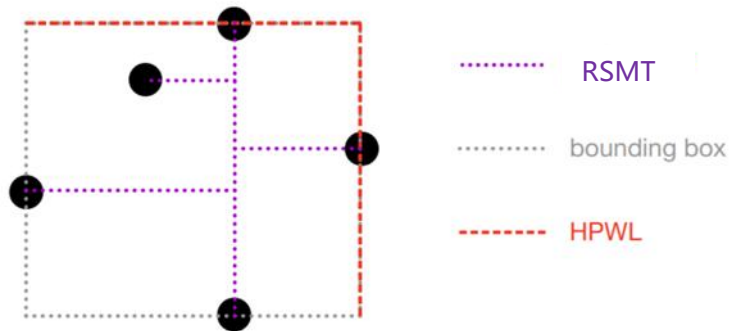
- v is cell location, $WL(v)$ is wirelength, $\rho_b(v)$ is the area density in $b \in B$, ρ_0 is density threshold



- The constrained optimization problem can be transformed into an unconstrained optimization problem by using relaxation technique.

Wirelength

- Wirelength Calculation



$$HPWL(e) = \max_{i,j \in e} |x_i - x_j| + \max_{i,j \in e} |y_i - y_j|$$

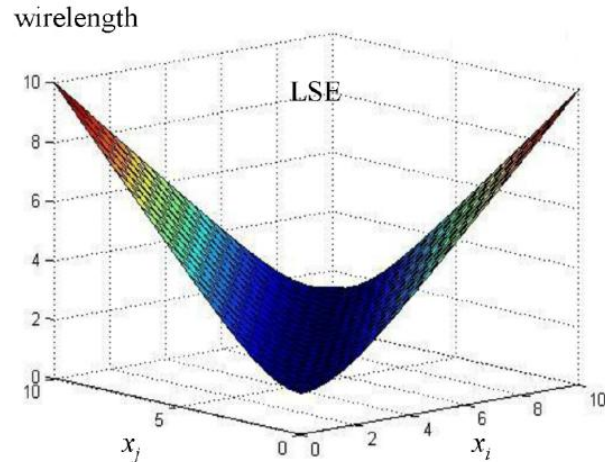
Wirelength Smoothing

■ Log-Sum-Exp(LSE)

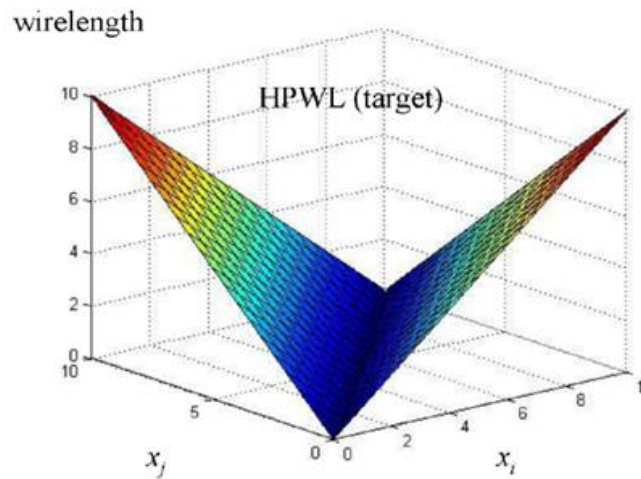
- 其中 γ 是平滑参数, 用于调节精确度, LSE模型误差的上界是, $\epsilon_{\text{LSE}}(e) \leq \gamma \log n$

■ Weighted-Average(WA)

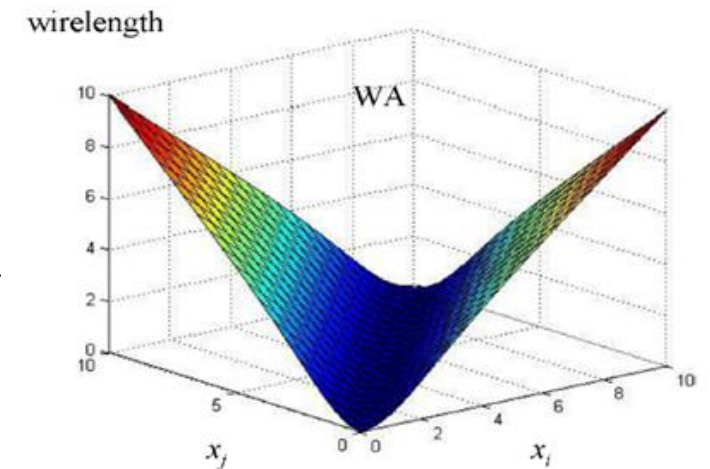
- 其中 γ 是平滑参数, 用于调节精确度, WA模型误差的上界是, $\epsilon_{\text{WA}}(e) \leq \frac{\gamma \Delta x}{1 + e^{\Delta x/n}}$



$$LSE_{ex} = \gamma \left(\ln \left(\sum_{i \in e} \exp \left(\frac{x_i}{\gamma} \right) \right) + \ln \left(\sum_{i \in e} \exp \left(\frac{-x_i}{\gamma} \right) \right) \right)$$



$$HPWL_{ex}(\mathbf{v}) = \max_{i,j \in e} |x_i - x_j|$$



$$WA_{ex} = \frac{\sum_{i \in e} x_i \exp \left(\frac{x_i}{\gamma} \right)}{\sum_{i \in e} \exp \left(\frac{x_i}{\gamma} \right)} - \frac{\sum_{i \in e} x_i \exp \left(\frac{-x_i}{\gamma} \right)}{\sum_{i \in e} \exp \left(\frac{-x_i}{\gamma} \right)}$$

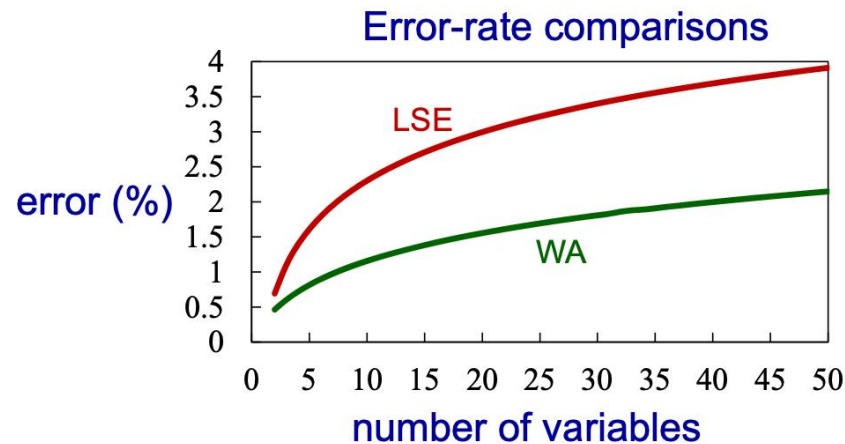
- Theorem: The estimation error bound of the WA model is

$$0 \leq \varepsilon_{WA}(\mathbf{x}_e) \leq \frac{\gamma(x_{\max} - x_{\min})}{1 + e^{(x_{\max} - x_{\min}) / n}}.$$

$\mathbf{x}_e = \{x_i | v_i \in e\}$: a set $\mathbf{x}$ of coordinates associated with net e

- Theorem: The **error upper bound** of the WA model is **smaller** than that of the LSE model:

$$\varepsilon_{WA}(\mathbf{x}_e) \leq \varepsilon_{LSE}(\mathbf{x}_e) = \gamma \ln n$$



- The WA model can achieve averagely **2% shorter total wirelength** than the LSE model

Wirelength Model	Wirelength	CPU Time
LSE	1.000	1.000
WA	0.980	1.066

- The results show that WA outperforms LSE consistently

Wirelength Smoothing

HPWL $W(\mathbf{x}, \mathbf{y}) = \sum_{\text{net } e} (\max_{v_i, v_j \in e} |x_i - x_j| + \max_{v_i, v_j \in e} |y_i - y_j|)$

quadratic $\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \gamma_{ij} [(x_i - x_j)^2 + (y_i - y_j)^2]$

Log-sum-exp $\gamma \sum_{e \in E} (\log \sum_{v_k \in e} \exp(x_k/\gamma) + \log \sum_{v_k \in e} \exp(-x_k/\gamma) + \log \sum_{v_k \in e} \exp(y_k/\gamma) + \log \sum_{v_k \in e} \exp(-y_k/\gamma))$

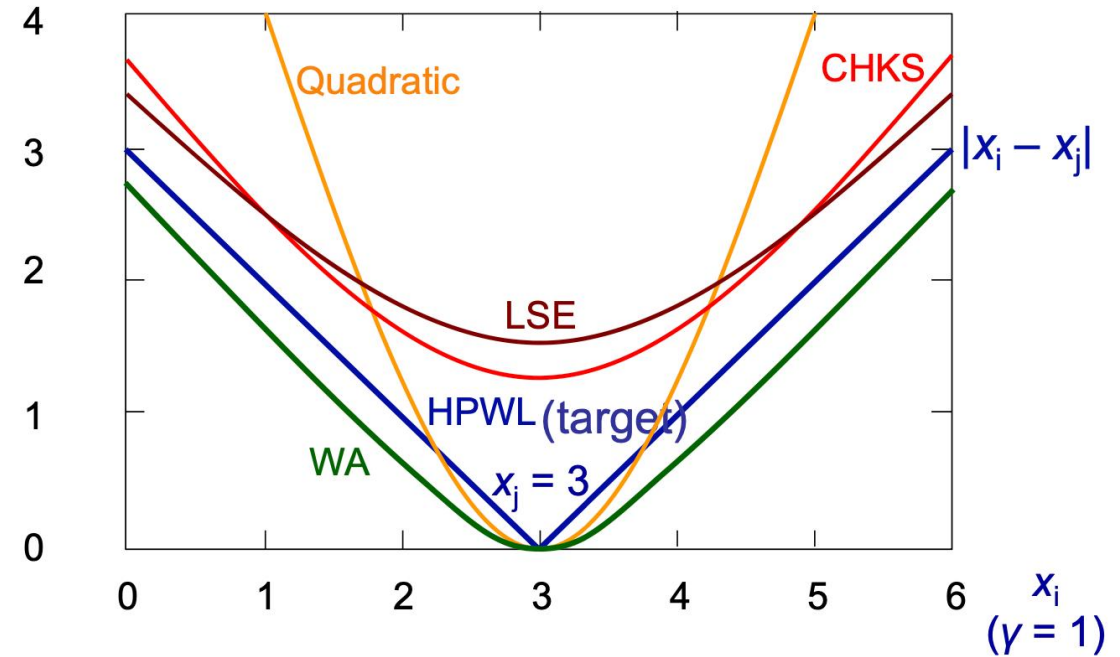
L_p-norm $\sum_{e \in E} ((\sum_{v_k \in e} x_k^p)^{\frac{1}{p}} - (\sum_{v_k \in e} x_k^{-p})^{-\frac{1}{p}} + (\sum_{v_k \in e} y_k^p)^{\frac{1}{p}} - (\sum_{v_k \in e} y_k^{-p})^{-\frac{1}{p}})$

CHKS $CHKS(x_1, x_2) = \frac{\sqrt{(x_1 - x_2)^2 + t^2} + x_1 + x_2}{2}$

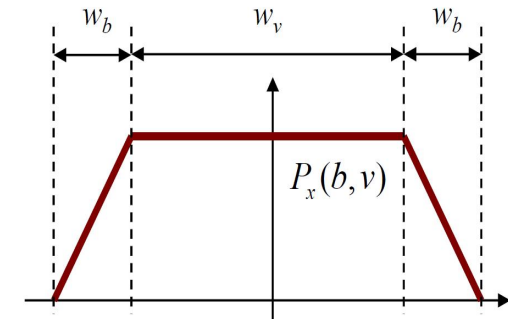
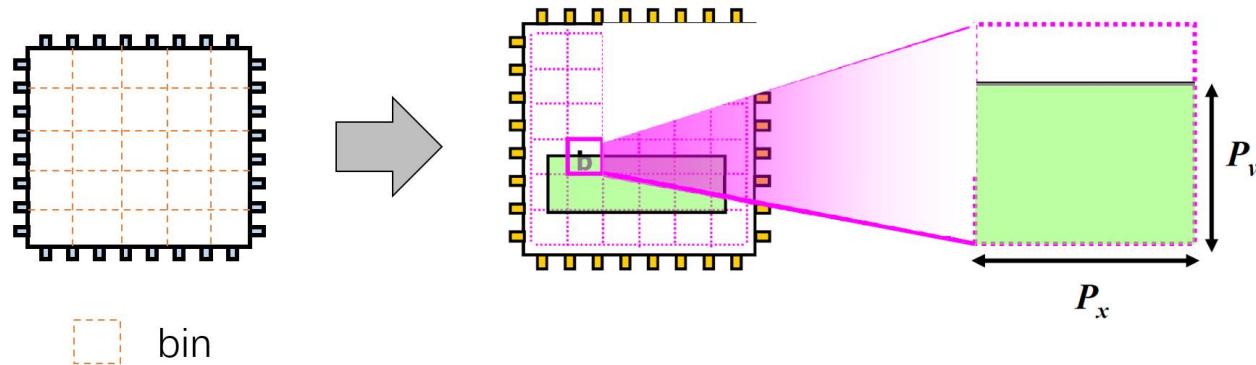
Weighted-average $\sum_{e \in E} \left(\frac{\sum_{v_i \in e} x_i \exp(x_i/\gamma)}{\sum_{v_i \in e} \exp(x_i/\gamma)} - \frac{\sum_{v_i \in e} x_i \exp(-x_i/\gamma)}{\sum_{v_i \in e} \exp(-x_i/\gamma)} + \frac{\sum_{v_i \in e} y_i \exp(y_i/\gamma)}{\sum_{v_i \in e} \exp(y_i/\gamma)} - \frac{\sum_{v_i \in e} y_i \exp(-y_i/\gamma)}{\sum_{v_i \in e} \exp(-y_i/\gamma)} \right)$

T.-Y. Chiang

wirelength Smooth, differential functions with 2 variables



- Density Calculation



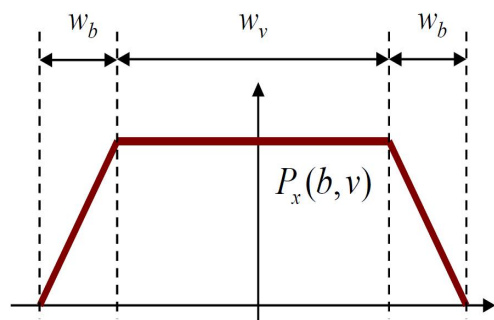
$$D(x, y) = \sum_{v \in V} P_x(b, v) P_y(b, v)$$

$$P_x(b, v) = \begin{cases} \frac{w_b + w_v}{2} - d_x & 0 \leq d_x \leq w_b - w_v \\ 0 & w_b - w_v \leq d_x \leq \frac{w_b + w_v}{2} \\ 0 & \frac{w_b + w_v}{2} \leq d_x \end{cases}$$

$d_x \triangleq |x_b - x_v|$

- **Bellshape Density**

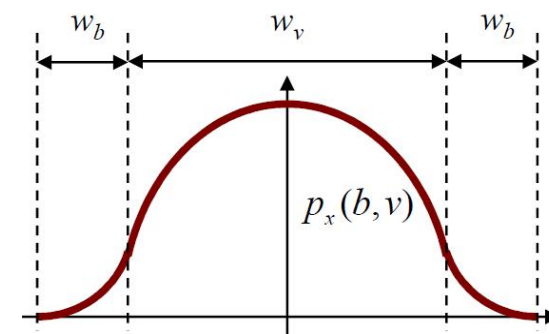
- 二次光滑化
- 高密度区域较陡，不利于布局优化



$$D(x, y) = \sum_{v \in V} P_x(b, v) P_y(b, v)$$

$$P_x(b, v) = \begin{cases} \frac{w_v}{w_b + w_v} - d_x & 0 \leq d_x \leq w_b - w_v \\ \frac{w_v + w_b}{2} - d_x & w_b - w_v \leq d_x \leq \frac{w_v + w_b}{2} \\ 0 & \frac{w_v + w_b}{2} \leq d_x \end{cases}$$

$$d_x = |x_b - x_v|$$



$$D'(x, y) = \sum_{v \in V} c_v P'_x(b, v) P'_y(b, v)$$

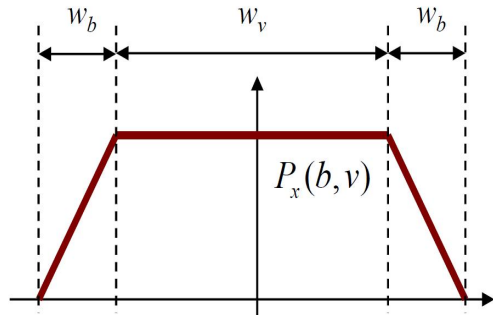
$$P'_x(b, v) = \begin{cases} 1 - ad_x^2, & 0 \leq d_x \leq \frac{w_v}{2} + w_b \\ b(d_x - 2w_b - 2w_v)^2, & \frac{w_v}{2} + w_b \leq d_x \leq \frac{w_v}{2} + 2w_b \\ 0, & \frac{w_v}{2} + 2w_b \leq d_x \end{cases}$$

$$a = 4 / ((w_v + 2w_b)(w_v + 4w_b))$$

$$b = 2 / (w_b(w_v + 4w_b))$$

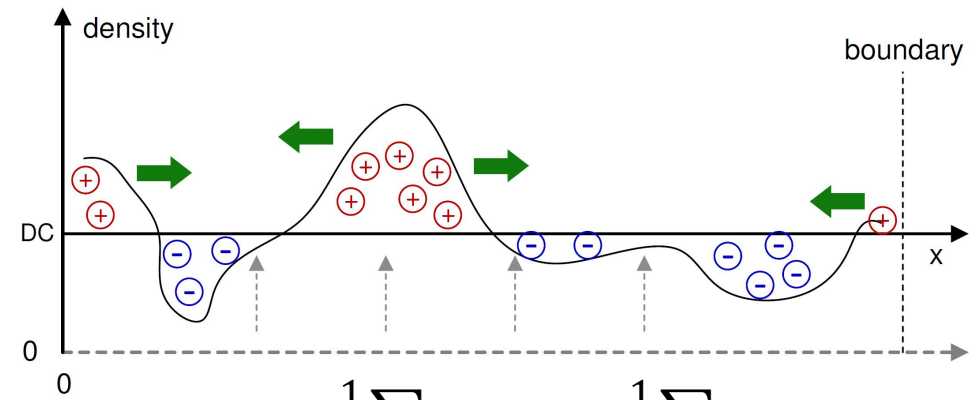
- Electronic Filed

- 电场能优化较平滑,
- 利于单元散开, 但不利于局部优化



$$D(x, y) = \sum_{v \in V} P_x(b, v) P_y(b, v)$$

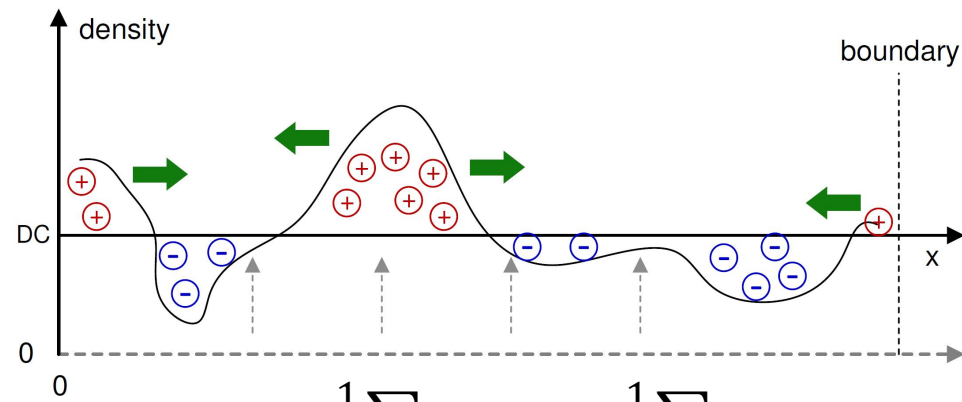
$$P_x(b, v) = \begin{cases} \frac{w_v}{w_b + w_v} - d_x & 0 \leq d_x \leq w_b - w_v \\ 0 & \frac{w_v + w_b}{2} \leq d_x \leq w_b + w_v \\ 0 & |x_b - x_v| \leq \frac{w_v + w_b}{2} \end{cases}$$



$$D(x, y) = \frac{1}{2} \sum_{v \in V} D_i(x, y) = \frac{1}{2} \sum_{v \in V} q_i \psi_i(x, y)$$

$$\begin{cases} \nabla \cdot \nabla \psi(x, y) = -\rho(x, y), \\ \hat{n} \cdot \psi(x, y) = 0, \quad (x, y) \in \partial R \\ \iint_R \rho(x, y) = \iint_R \psi(x, y) = 0. \end{cases}$$

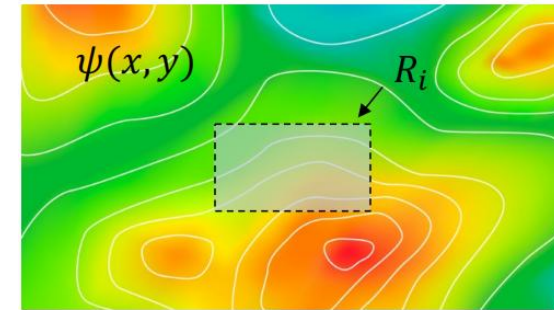
- Electronic Filed
 - 离散余弦变换求解
 - 离散化



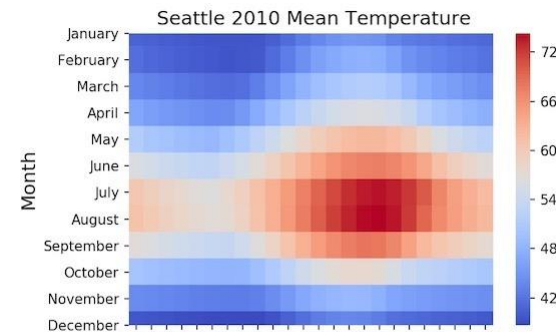
$$D(x, y) = \frac{1}{2} \sum_{v \in V} D_i(x, y) = \frac{1}{2} \sum_{v \in V} q_i \psi_i(x, y)$$

$$\begin{cases} \nabla \cdot \nabla \psi(x, y) = -\rho(x, y), \\ \hat{n} \cdot \psi(x, y) = \mathbf{0}, \quad (x, y) \in \partial R \\ \iint_R \rho(x, y) = \iint_R \psi(x, y) = 0. \end{cases}$$

$$\psi(x, y) = \sum_{u=0}^K \sum_{p=0}^K a_{up} \cos\left(\frac{u\pi}{W}x\right) \cos\left(\frac{p\pi}{H}y\right)$$



$$\psi_i q_i = \iint_{R_i} \psi(x, y) \cdot 1 \, dx dy$$



$$\psi_k q_k = \iint_{R_k} \psi(x, y) \, dx dy \approx \sum_{v_i \cap B} \text{Area}(v_k \cap \text{bin}_{i,j}) \psi(i, j)$$

Modern Global Placement

■ Min Wirelength Model

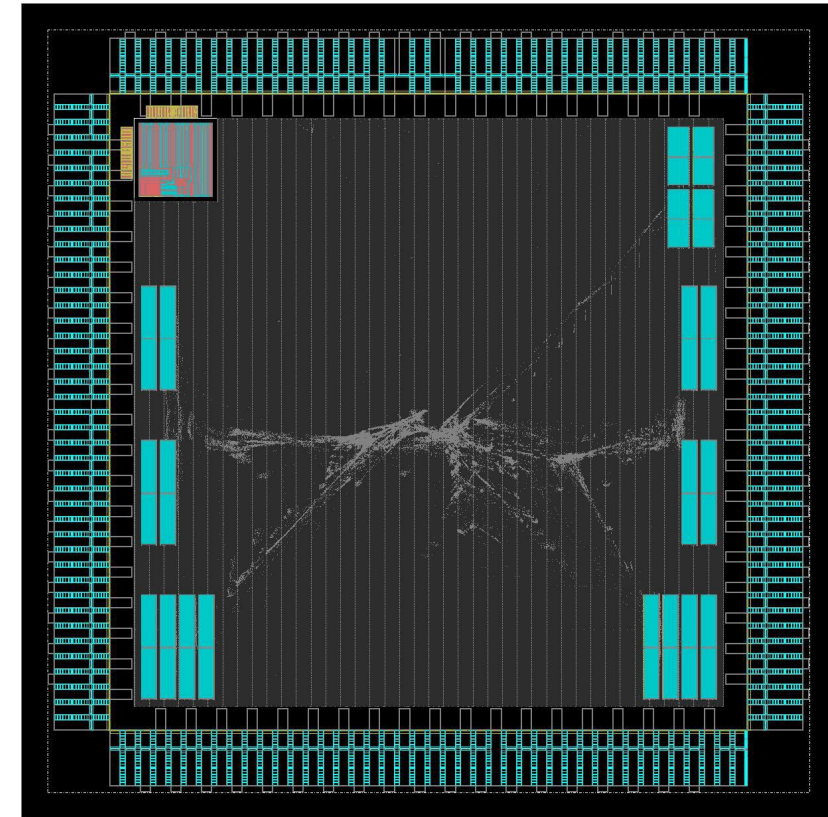
$$\begin{aligned} \min_v \quad & WL(v) \\ \text{s. t.} \quad & \rho_b(v) \leq \rho_0, \quad \forall b \in B \end{aligned}$$

where v is cell location, $WL(v)$ is wirelength, $\rho_b(v)$ is the area density in $b \in B$, ρ_0 is density threshold.

$$WL(v) \left\{ \begin{aligned} HPWL_{ex}(v) &= \max_{i,j \in e} |x_i - x_j| \\ LSE_{ex} &= \gamma \left(\ln \left(\sum_{i \in e} \exp \left(\frac{x_i}{\gamma} \right) \right) + \ln \left(\sum_{i \in e} \exp \left(\frac{-x_i}{\gamma} \right) \right) \right) \end{aligned} \right.$$

$$\rho_b(v) \left\{ \begin{aligned} D(v) &= \frac{1}{2} \sum_{v \in V} D_i(x, y) = \frac{1}{2} \sum_{v \in V} q_i \psi_i(x, y) \\ \begin{cases} \nabla \cdot \nabla \psi(x, y) = -\rho(x, y), \\ \hat{n} \cdot \psi(x, y) = 0, \quad (x, y) \in \partial R \end{cases} \\ \iint_R \rho(x, y) = \iint_R \psi(x, y) = 0. \end{aligned} \right.$$

$$\min_v f(v) = WL(v) + \lambda \sum_{b \in B} \rho_b(v)$$



- Nesterov Method or Conjugate Gradient

梯度下降法求解

$$P_1: \min_v f(v) = W(v) + \lambda^T N(v)$$

$$\nabla f(v) = \nabla W(v) + \lambda^T \nabla N(v)$$

• 共轭梯度法

- 一次迭代中 CG 方法的详细信息如算法 1 所示。Polak-Ribiere 方法用于更新 β_k 以与之前的搜索方向相关，当共轭性丢失时， β_k 重置为零。根据 β_k 计算梯度方向。使用黄金分割搜索（GSS）求出步长。当前迭代的新解在第 5 行计算并用作下一次迭代的初始解，而CG会在经过多次这样的迭代后收敛迭代。CG 的目标是局部二次函数的优化。f 越接近二次形式，CG 收敛得越快。

• Nesterov法

- 与 CG 方法类似，Nesterov 的方法仅需要一阶梯度和相对于问题大小的线性内存成本。在算法2中，u为主解，v 为参考解。第二行通过计算，Lipschitz constant 获得步长。新解 u_{k+1} 在第 3 行基于初始参考解 v_k 进行更新。新的优化参数 a_{k+1} 在第4行更新，而新的参考解 v_{k+1} 在第5行基于解u和参数a更新。Nesterov方法是一种在凸优化问题中常用的优化算法，通过引入动量来加速参数更新。它的主要优点是避免了传统梯度下降算法的局部最小值问题，并能够在优化过程中逐渐减小更新步长。

ALGORITHM 1: CG-Solver at k th iteration

Input: initial solution $\mathbf{v}_k$

objective function $f_k = f(\mathbf{v}_k)$ maximal and minimal search interval α_k^{max} and α_k^{min}

Output: local optimal solution $\mathbf{v}_{k+1}$

1: gradient vector $\nabla f_k = \nabla f(\mathbf{v}_k)$

2: Polak-Ribiere parameter $\beta_k = \max \left\{ \frac{\nabla f_k^T (\nabla f_k - \nabla f_{k-1})}{\|\nabla f_{k-1}\|^2}, 0 \right\}$

3: search direction $\mathbf{d}_k = -\nabla f_k + \beta_k \mathbf{d}_{k-1}$

4: steplength $\alpha_k = GSS(\mathbf{v}_k, f_k, \mathbf{d}_k, \alpha_k^{max}, \alpha_k^{min})$

5: new solution $\mathbf{v}_{k+1} = \mathbf{v}_k + \alpha_k \mathbf{d}_k$

6: **return** $\mathbf{v}_{k+1}$

Algorithm 2. Nesterov-Solver at k th iteration

Require: major solution $\mathbf{u}_k$, reference solution $\mathbf{v}_k$, optimization parameter a_k and objective function $f_k = f(\mathbf{y}_k)$.

Ensure: new solutions $\mathbf{u}_{k+1}$ and $\mathbf{v}_{k+1}$

1: gradient vector $\nabla f_k = \nabla f(\mathbf{v}_k)$

2: steplength $\alpha_k = \arg \max_{\alpha} \{f_k - f(\mathbf{v}_k - \alpha \nabla f_k) \geq 0.5 \alpha \|\nabla f_k\|^2\}$

3: new solution $\mathbf{u}_{k+1} = \mathbf{v}_k - \alpha_k \nabla f_k$

4: parameter update $a_{k+1} = \left(1 + \sqrt{4a_k^2 + 1}\right) / 2$

5: new reference solution $\mathbf{v}_{k+1} = \mathbf{u}_{k+1} + (a_k - 1)(\mathbf{u}_{k+1} - \mathbf{u}_k) / a_{k+1}$

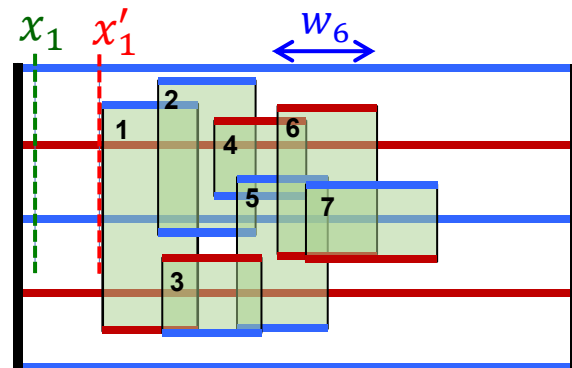
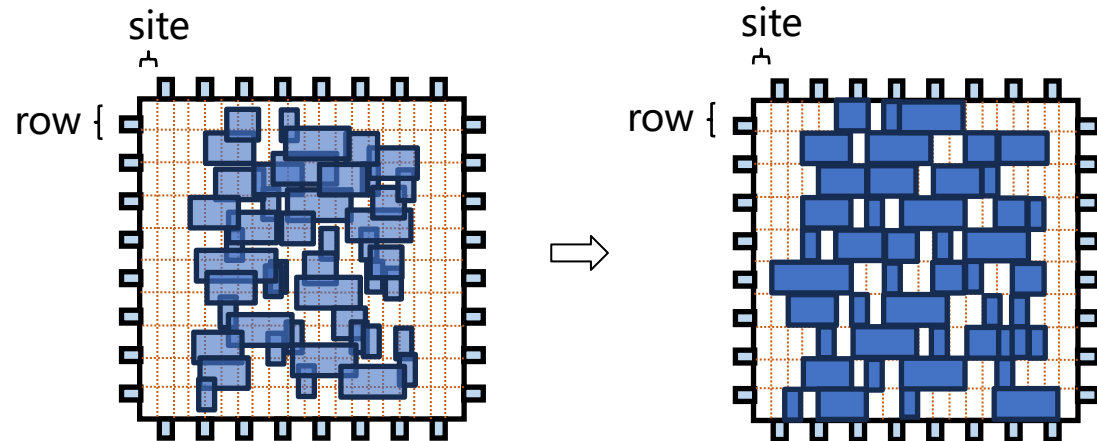
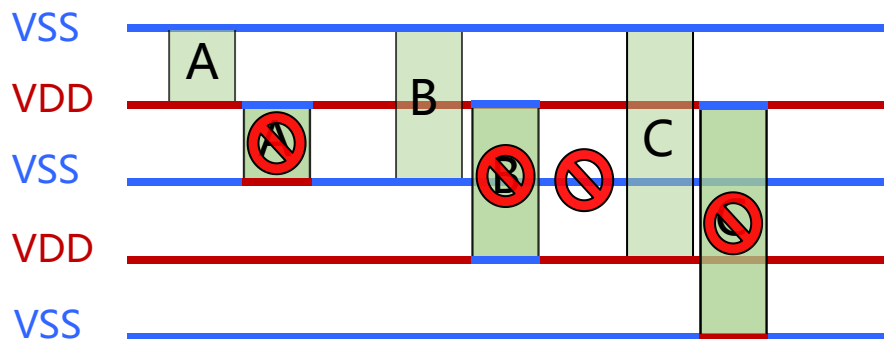
6: **return** $\mathbf{u}_{k+1}$

- 01** Introduction
- 02** Placement Techniques
- 03** Global Placement
- 04** **Detail Placement**
- 05** Timing Driven Placement
- 06** Congestion Driven Placement

合法化

● 问题描述: 给定全局布局的结果, 进行合法化, 使得单元移动量最小, 同时下面约束满足

- 1) 单元需放置在芯片区域内;
- 2) 单元之间不能互相重叠;
- 3) 单元需对齐到正确的行 (VDD/VSS需对齐)



● Tetris

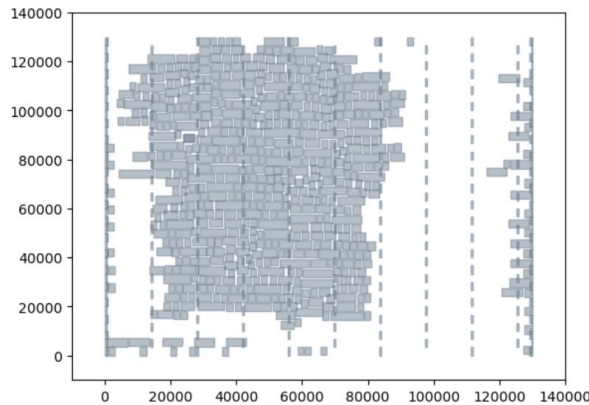
- 基于贪心算法，把单元按照x左边从左往右排序，依次为每个单元找一个当前最好的位置摆放

● Abacus

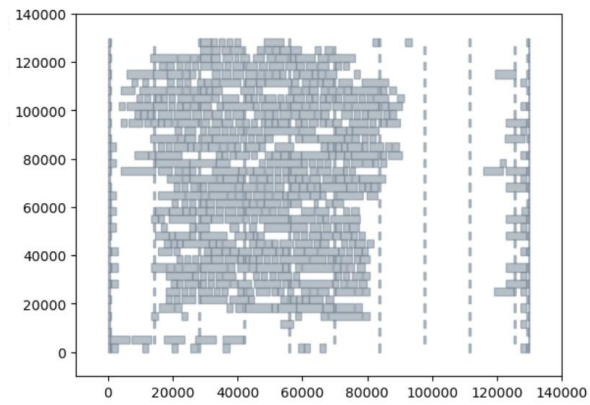
- 基于动态规划，在Tetris的基础上会为前面已经摆放的单元的整体最好的位置

结果对比

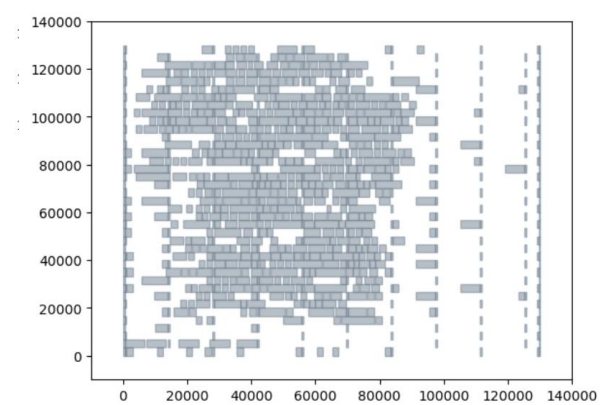
part metrics	abacus	tetris
global placement HPWL	10127910	10127910
legalization HPWL	10426323	13168231
detail placement HPWL	9901517	10928985
detail placement STWL	10637190	11674987
maximum STWL	431085	415325
total movement	795829	8705103
maximum movement	5684	218214
average congestion	0.8215	0.8134
total overflow	49	49
peak bin density	1	1
legalization runtime (s)	0.0667	0.0064



Input



Abacus

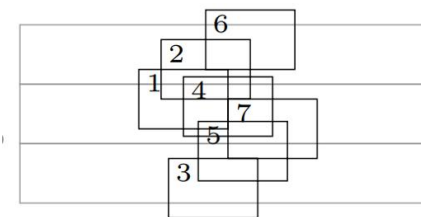


Tetris

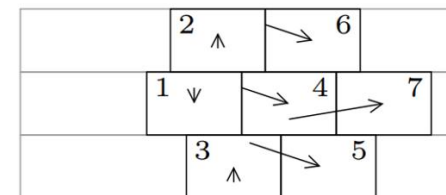
- iEDA实践: <https://ieda.oscc.cc/train/practice/algorithms/a1.html>

■ 基础算法：

- 首先对所有单元按照**横坐标顺序**进行排列，按顺序在每一行选取**最左端**的一个**空白区域**作为候选空间。在所有候选空间中挑出**最近**的一个，并将该单元放入。



(a) Global Placement



(b) Tetris. Total Movement: 3.752

■ 变体：

- **行密度拓展**：大于某个密度的行不选取
- **行选取的拓展**：距离开始行比较近的地方
- **局部拓展**：将整个布局区域分为k份,在局部内做合法化。

优点：**速度**很快。

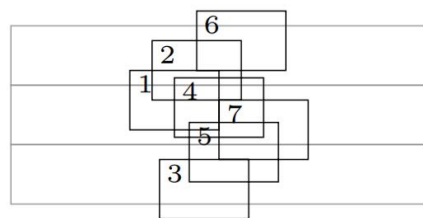
缺点：质量不**稳定**，**局部**获得良好的结果，但是**整体**结果较差。

■ Abacus:

- 首先对所有单元按照从**横坐标顺序**进行排列，依据当前**总体局面**，逐个为每个标准单元选取**最合适**的行，允许**行内**的**适当调整**。

优点：**质量**相比Tetris有较大提升。

缺点：质量不够**稳定**，依赖遍历的**顺序**，当存在较大的**宽度差**的时候，结果**显著**变差，速度变慢。

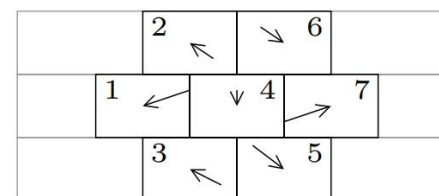


(a) Global Placement

Algorithm 1: Our legalization approach “Abacus”

```

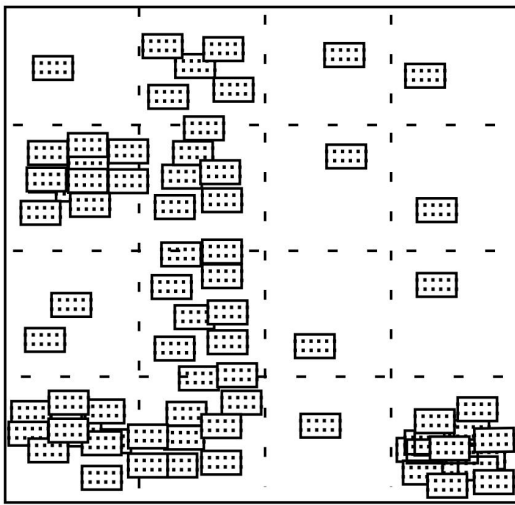
1 Sort cells according to x-position;
2 foreach cell  $i$  do
3    $C_{best} \leftarrow \infty$ ;
4   foreach row  $r$  do
5     Insert cell  $i$  into row  $r$ ;
6     PlaceRow  $r$  (trial);
7     Determine cost  $c$ ;
8     if  $c < C_{best}$  then  $C_{best} = c$ ,  $r_{best} = r$ ;
9     Remove cell  $i$  from row  $r$ ;
10  end
11  Insert Cell  $i$  to row  $r_{best}$ ;
12  PlaceRow  $r_{best}$  (final);
13 end
    
```



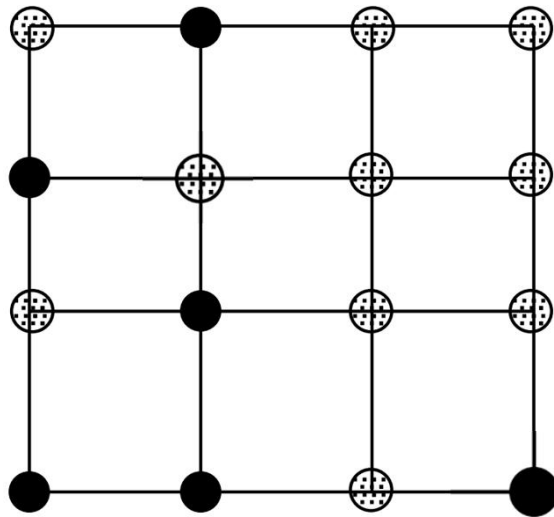
(c) Abacus. Total Movement:
2.915

■ 建图:

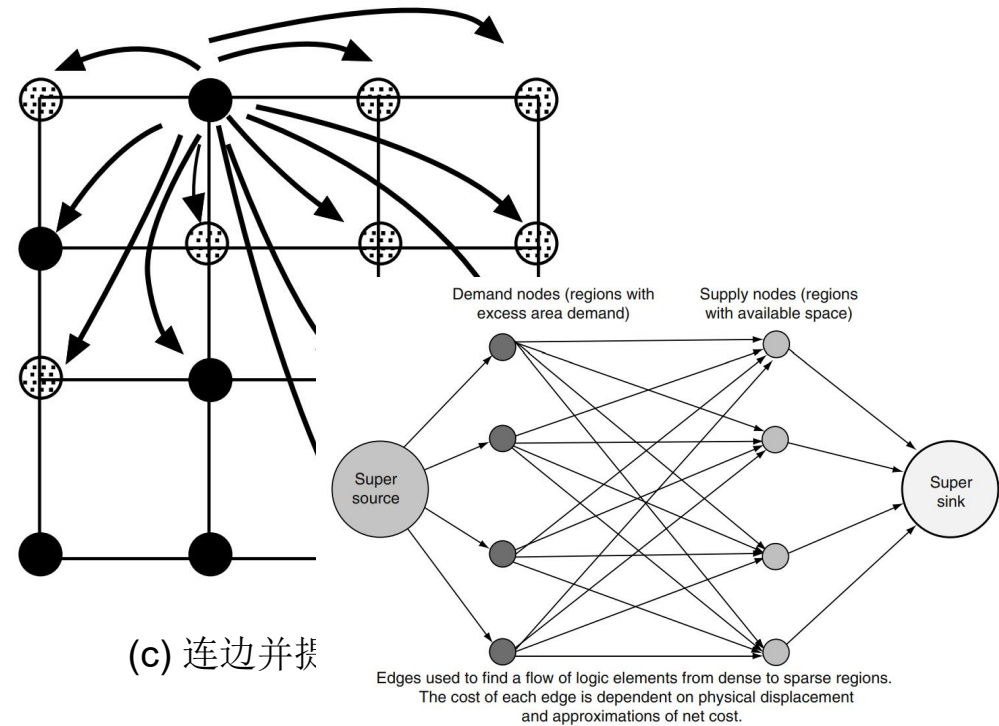
- 对整个版图划分格子，对格子之间进行连边，根据格子中的面积进行供给点和需求点分析。
为供给点和需求点提供流量



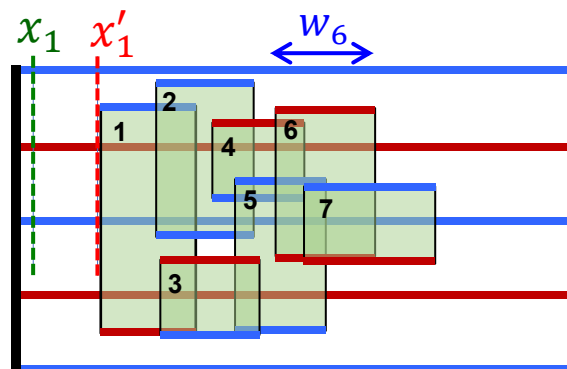
(a) 划分网格



(b) 连边，黑色为供给点，白色为需求点



(c) 连边并找



- 二次规划 (QP)

- 将合法化问题描述成二次规划问题

$$\begin{aligned} \min \quad & \frac{1}{2} \sum_{i=0}^n (x_i - x'_i)^2 \\ \text{s. t.} \quad & x_j - x_l \geq w_l, \text{ if } x'_j \geq x'_l, \quad \forall \text{ adjacent cells } j \text{ and } l \text{ in the same row;} \\ & x \geq 0. \end{aligned}$$

Placement-0306-带有障碍的标准单元合法化

■ 问题描述

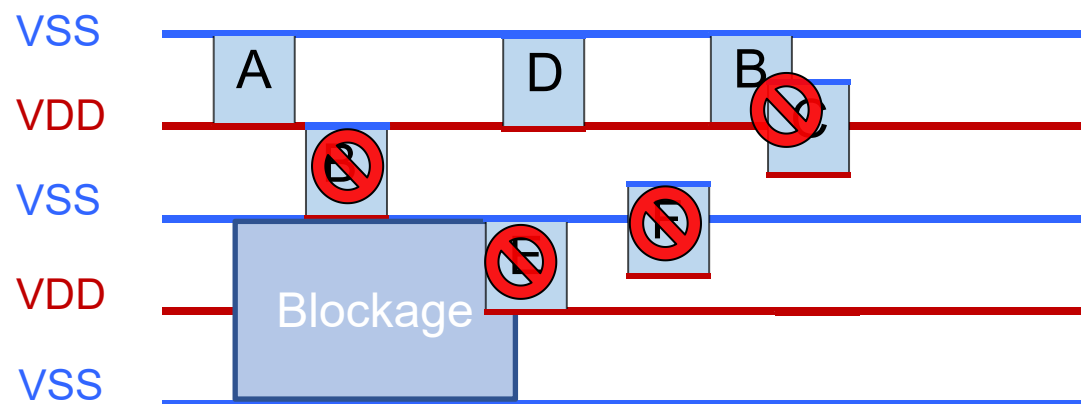
一个SoC级的设计可能包含一些IP，被当作**宏模块预先放置**在了版图之上，合法化阶段要保证标准单元与宏模块没有重叠。

■ 目标：

- 最小化全局移动量

■ 约束：

- 单元在芯片**区域内**
- 单元间**无重叠**
- 单元对齐**VDD/VSS**
- 单元与Block之间没有重叠



■ 算法设计挑战：

- 放置区域由于Fixed-blockage的影响，放置结果可能会变差。

■ 问题描述

- 随着先进的技术不断发展，布局区域约束为合法化问题带来巨大挑战。
- 原有矩形布局区域可能会变成一个或多个联通区域构成的图形，原有的一些合法化方法不得不重新考虑区域约束。

■ 目标：

- 最小化全局移动量

■ 约束：

- 单元在布局**区域内**
- 单元间**无重叠**
- 单元**VDD/VSS**
- 单元位于行上的placement sites

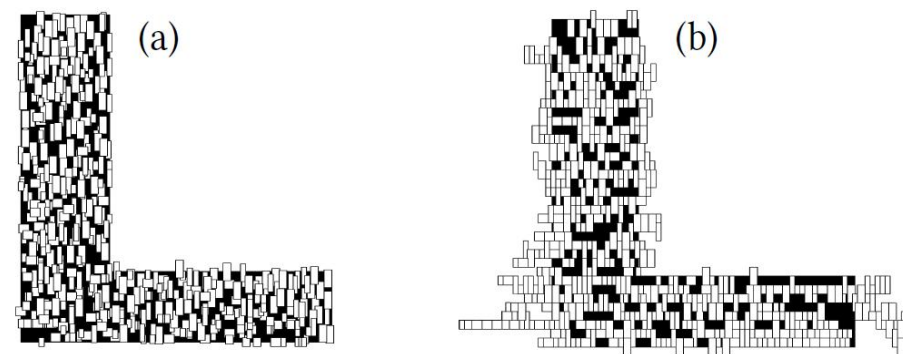
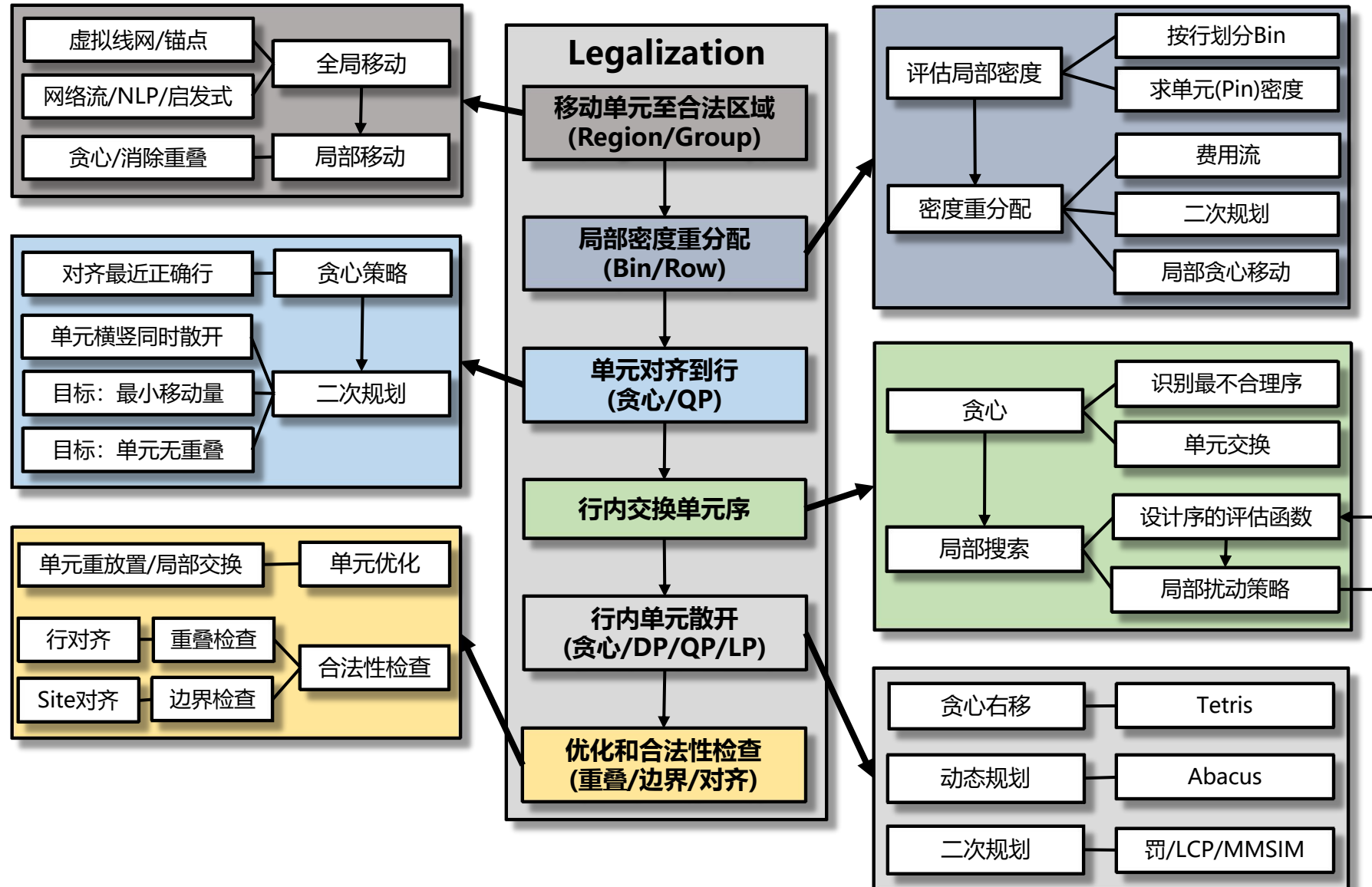


Figure 2: Examples of LCP based multi-deck standard cell legalization. (a) global placement result with a fence region. (b) faulty legalization result obtained using a Linear Complementary Problem with a fence region.

■ 算法设计挑战：

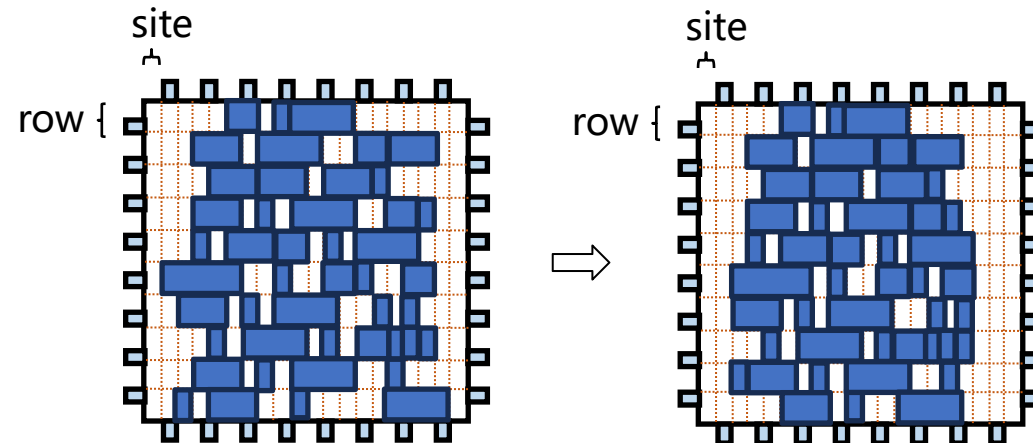
- 区域约束加入之后，不得不考虑在一些行无法放置的问题，需要在合适的情况下进行行上的重排列。

Legalization

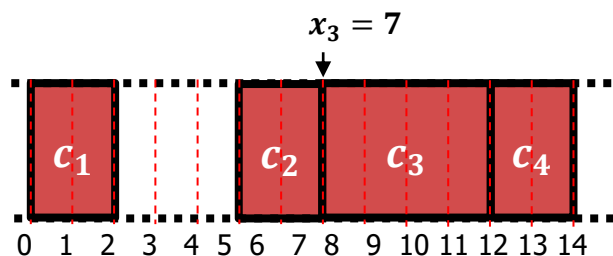


- 通过局部交互单元，实现指标更优化

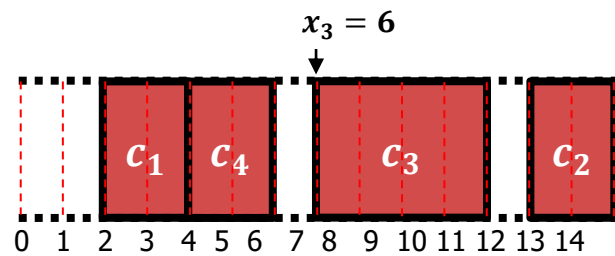
- 指标：线长，时序，拥塞，功耗
- 策略：行内移动，行内交互，行间交换，



- 行内移动例子

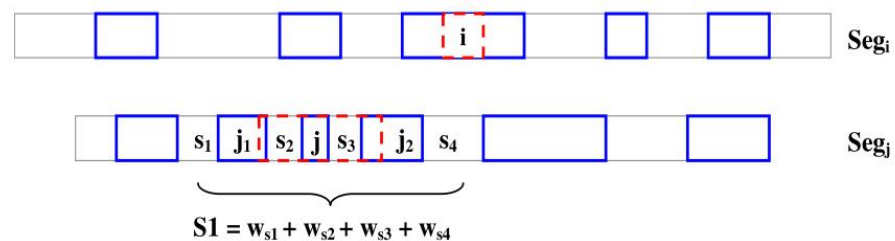


(a) Initial row placement



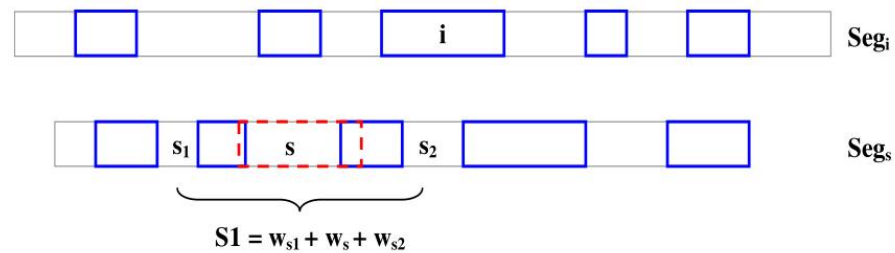
(b) An example of final solution

● 跨行交换



Penalty on swapping two cells i and j :

$$P1 = ((w_i - w_j) - (s_2 + s_3)) \times wt1$$

$$P2 = ((w_i - w_j) - S1) \times wt2$$


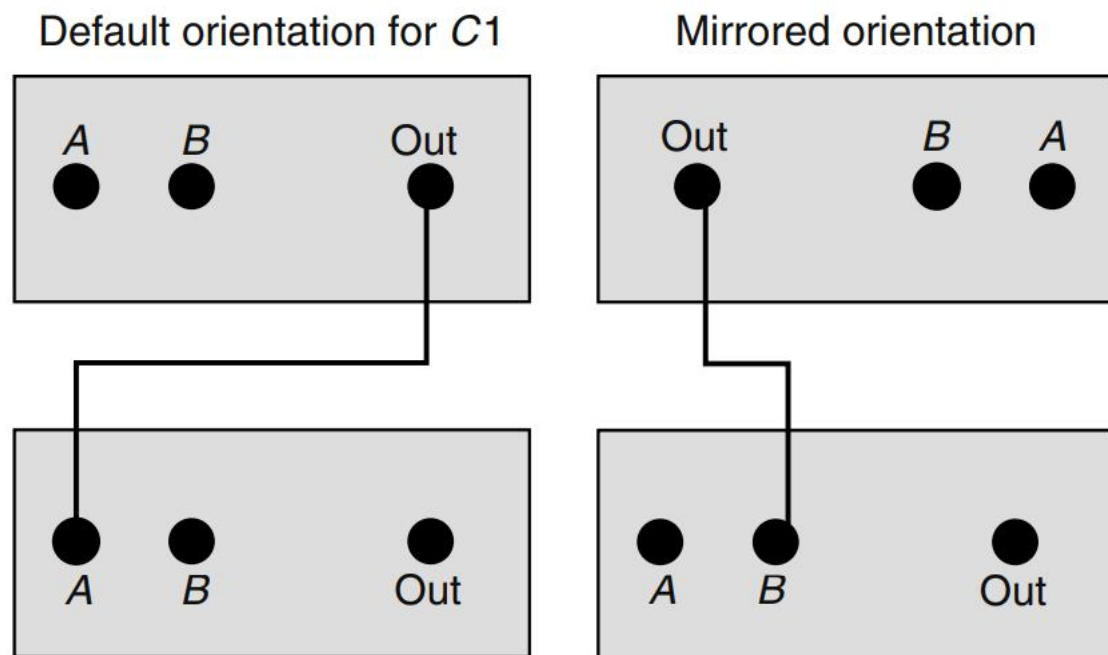
Penalty on swapping a cell i with a space s :

$$P1 = (w_i - s) \times wt1$$

$$P2 = (w_i - S1) \times wt2$$

详细布局

- 由于标准单元上的**Pin的位置**不同，当标准单元在水平方向进行**镜像对称**时，所得**线网**也是不同的，因此会对线长造成一定影响。



Optimal region for detailed placement *iEDA*

- Optimal region = target region for moving a cell in detailed placement
- Given by the x and y medians of all nets
 - Given nets $j=1\dots n$, with bounding box coordinates $x_j(l)$, $x_j(r)$, $y_j(l)$, $y_j(r)$, order all x' s and find the median; order all y' s and find the median

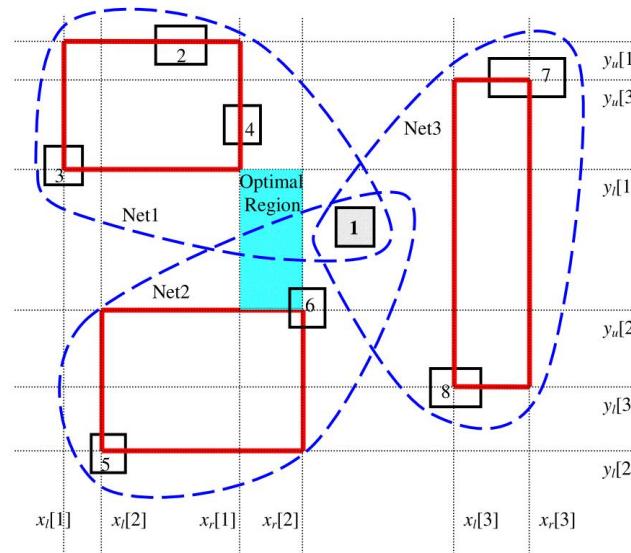
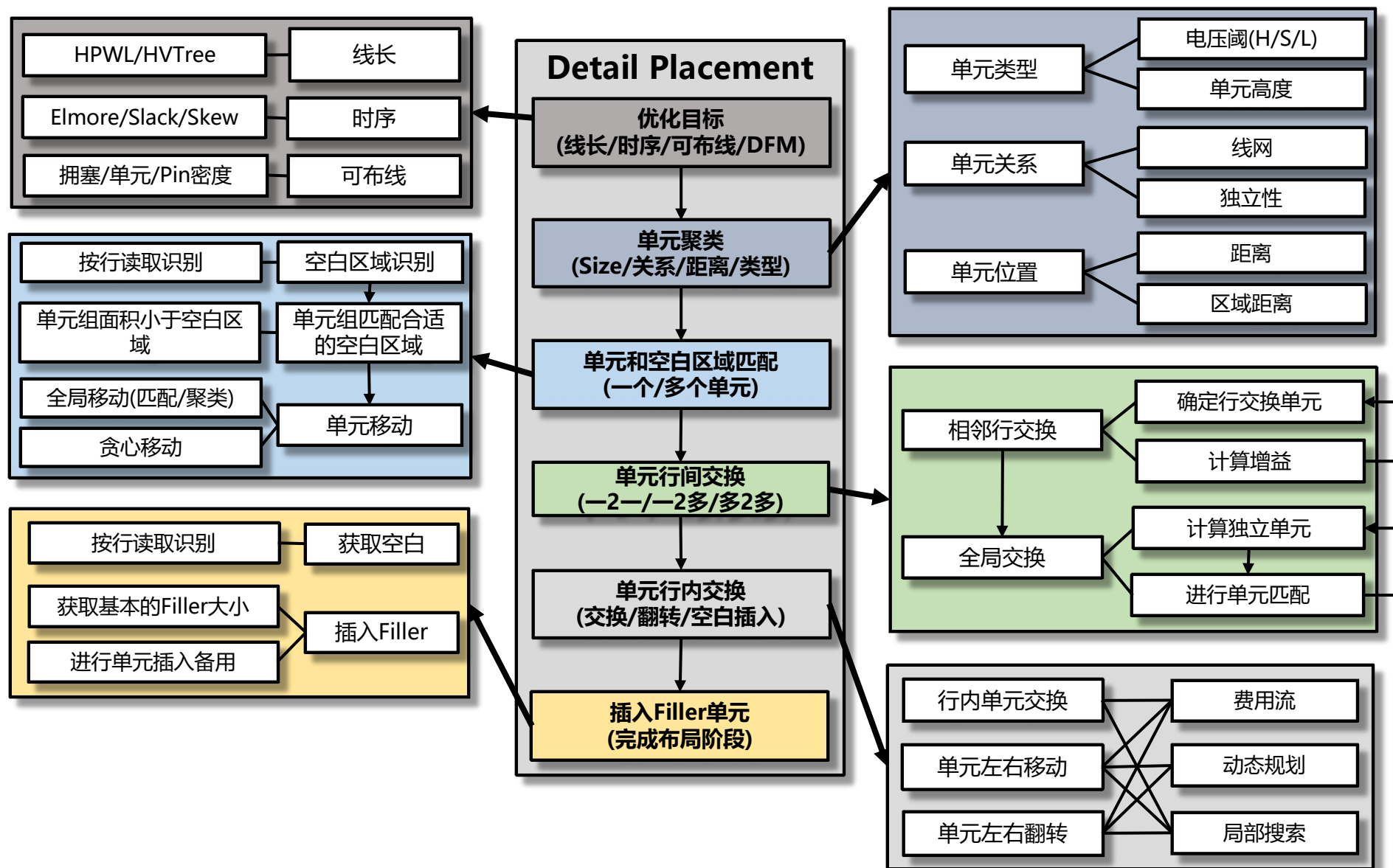


Fig. 2. Optimal Region.

Detailed Placement



- 01** Introduction
- 02** Placement Techniques
- 03** Global Placement
- 04** Detail Placement
- 05** **Timing Driven Placement**
- 06** Congestion Driven Placement

Modern Global Placement Model

- **P₀-线长**: 考虑线长目标和单元密度约束模型

- v 是单元坐标变量, $WL(v)$ 是线长函数, $\rho_b(v)$ 是Bin $b \in B$ 内单元的面积密度, ρ_0 是给定密度阈值

$$\begin{aligned} \min_v \quad & WL(v) \\ \text{s.t.} \quad & \rho_b(v) \leq \rho_0, \quad \forall b \in B \end{aligned}$$

- **P₁-时序**: 考虑线长、时序目标和单元密度、时序约束优化模型

- $T(v)$ 是关于坐标的时序函数, λ_1 是线长和时序权衡参数; $t_c(v)$ 是单元 $c \in C$ 的Slack, t_0 是给定Slack阈值

$$\begin{aligned} \min_v \quad & WL(v) + \lambda_1 T(v) \\ \text{s.t.} \quad & \rho_b(v) \leq \rho_0, \quad \forall b \in B \\ & t_c(v) \leq t_0, \quad \forall c \in C \end{aligned}$$

- **P₂-拥塞**: 考虑线长、Congestion目标和单元密度、Pin密度约束优化模型

- $R(v)$ 是关于坐标的Congestion函数, λ_2 是线长和Congestion权衡参数; $p_b(v)$ 是Bin $b \in B$ 内pin的密度函数, p_0 是给定pin密度阈值

$$\begin{aligned} \min_v \quad & WL(v) + \lambda_2 R(v) \\ \text{s.t.} \quad & \rho_b(v) \leq \rho_0, \quad \forall b \in B \\ & p_b(v) \leq p_0, \quad \forall b \in B \end{aligned}$$

- **P₃-区域**: 考虑线长目标和单元密度、Fence/Region约束优化模型

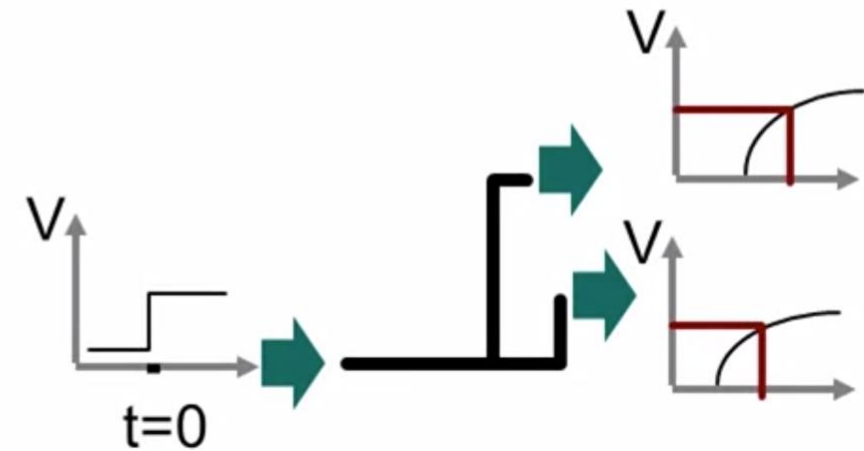
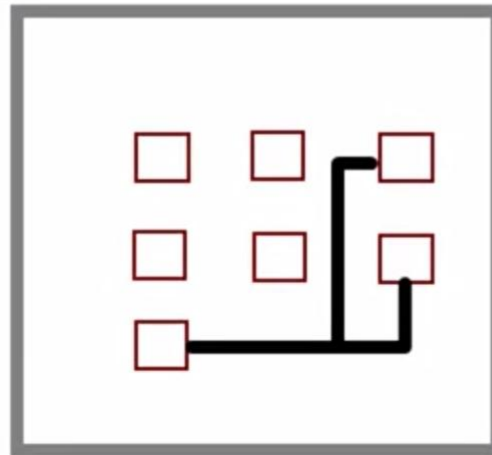
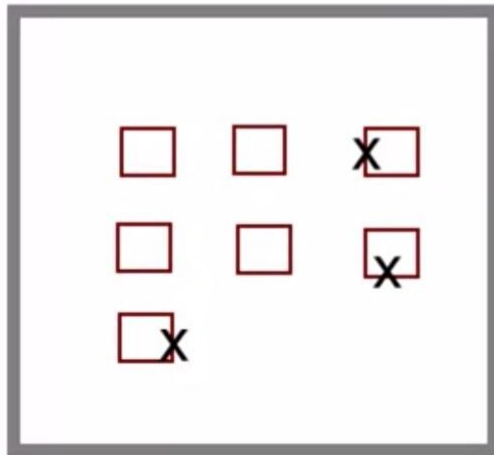
- $f(v)$ 是单元的Fence/Region约束函数, f_0 是给定Fence/Region阈值

$$\begin{aligned} \min_v \quad & WL(v) \\ \text{s.t.} \quad & \rho_b(v) \leq \rho_0, \quad \forall b \in B \\ & f(v) \leq f_0 \end{aligned}$$

Interconnect Delay

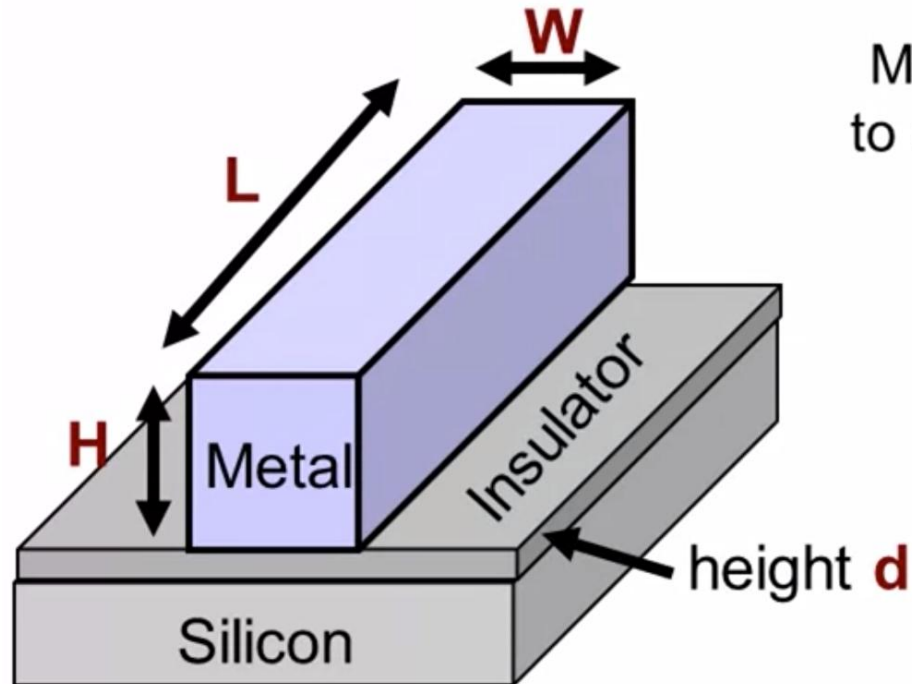
- **The problem**

- You **place the logic** it puts the pins at a certain distance apart
- You **route the wires**, each wire has an input-to-output delay
- **Where** does the delay come from? How accurately can we **predict** this delay?
- How efficiently can we **model** this delay for use in layout or synthesis or STA?



Wire Resistance

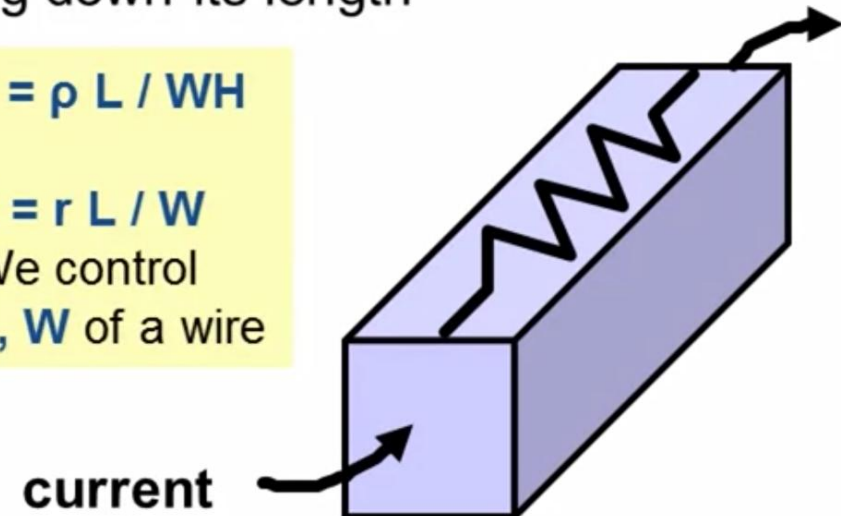
- Most popular interconnect model used in layout applications
- First: **Interconnect** → **Circuit**



Metal wire has **resistance = R**
to current flowing down its length

Physics: $R = \rho L / WH$

ASIC: $R = r L / W$
We control
 L, W of a wire

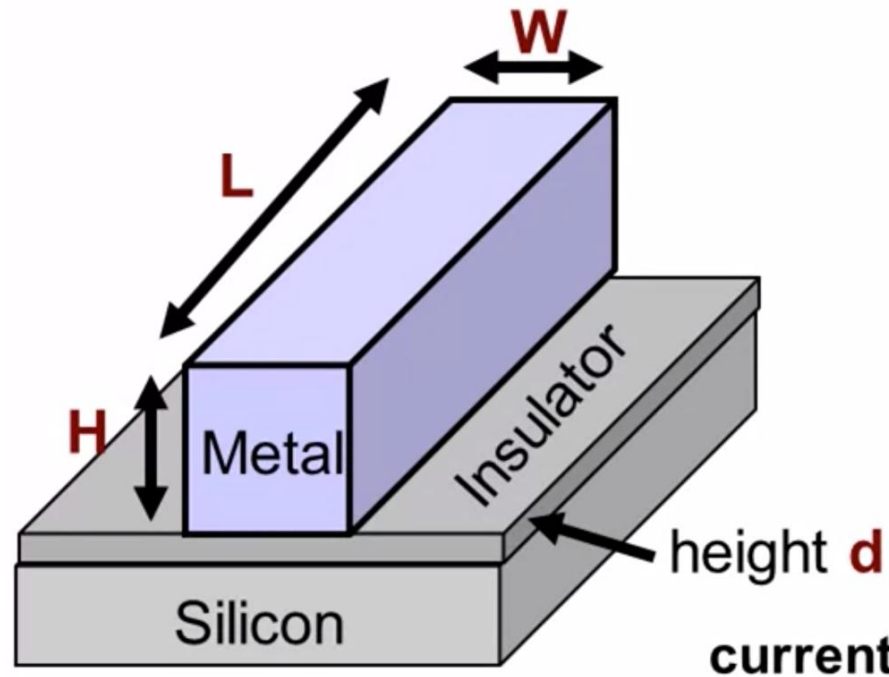


Wire Capacitance

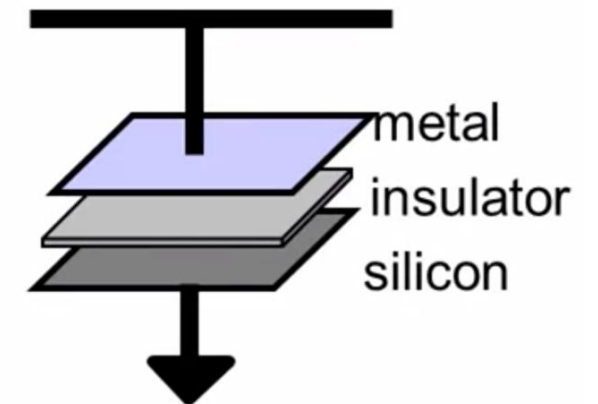
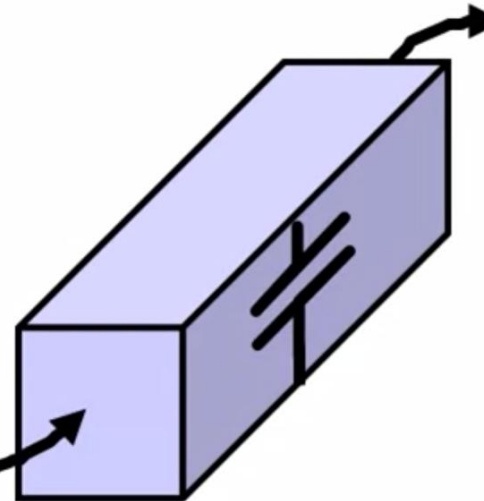
Metal wire has **capacitance** to silicon substrate, with insulator between

Physics: $C = \epsilon WL / d$

ASIC: $C = c WL$
We can control
 W, L of wire

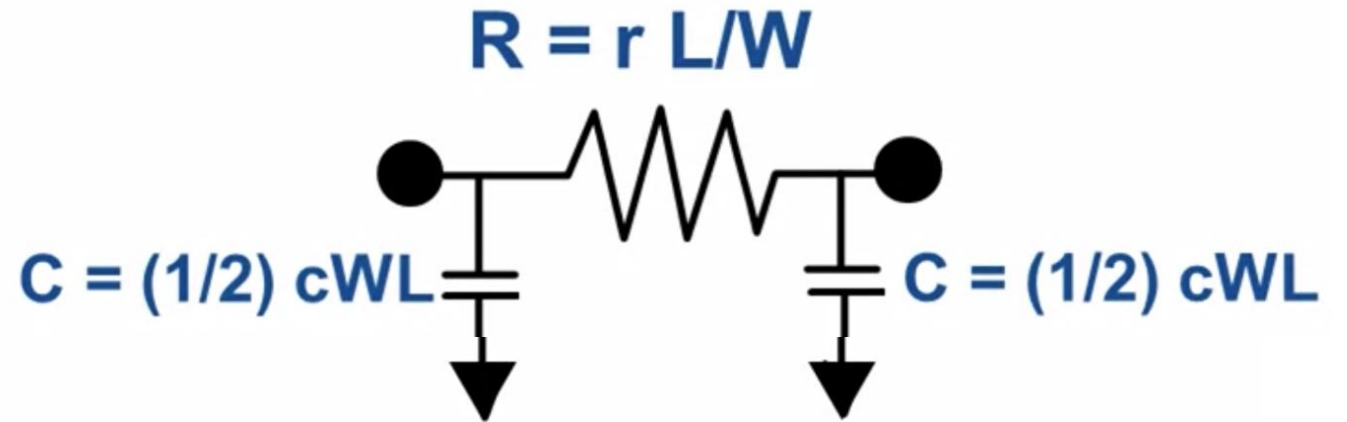
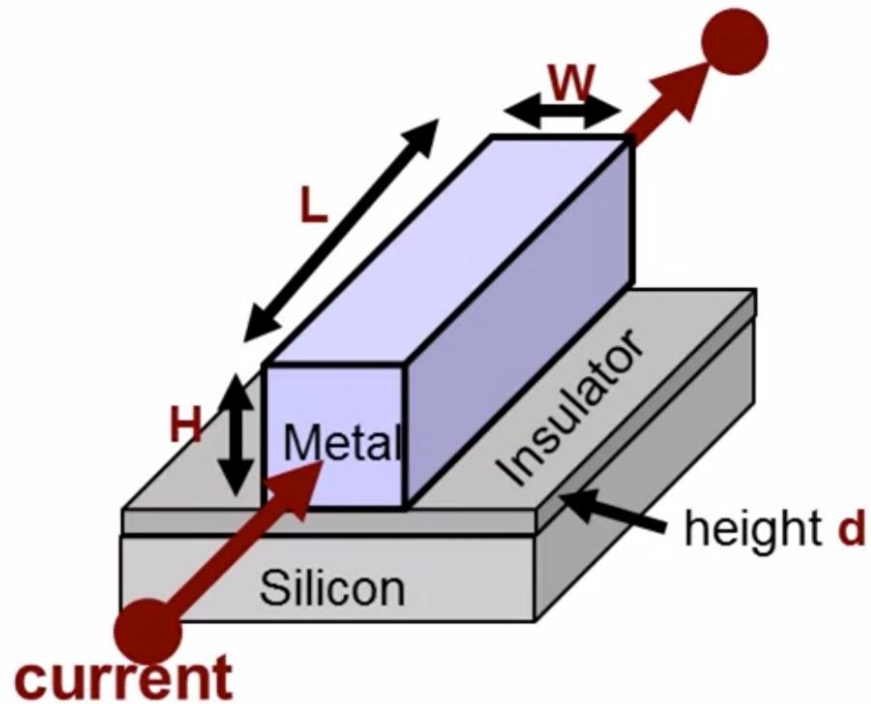


current



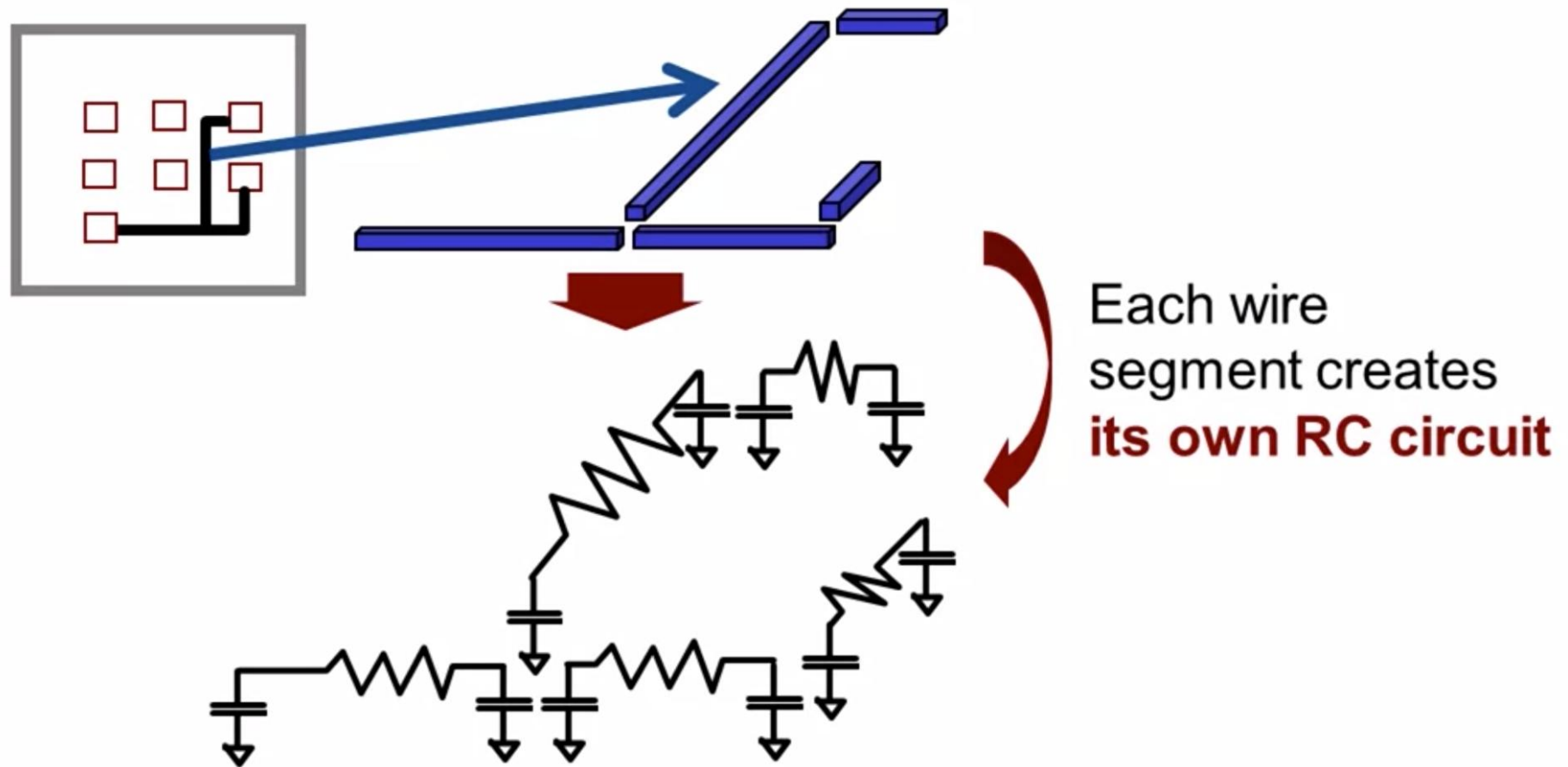
π Model

- **Typical circuit model: π model (“pi” model)**
 - Accounts for the resistance **R** and the capacitance **C** of wire segment
 - Symmetric (note: **split** capacitance in two halves); small model, only need **2** numbers



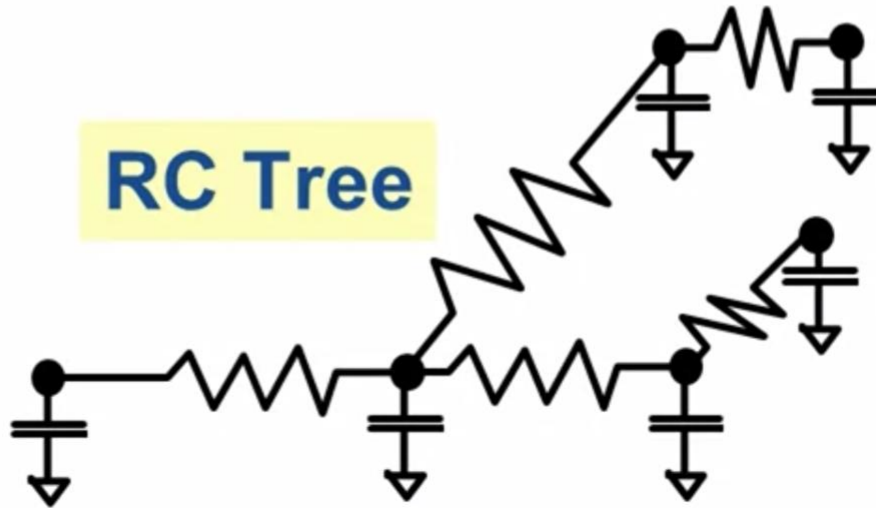
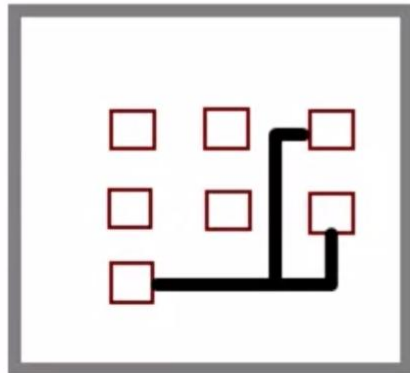
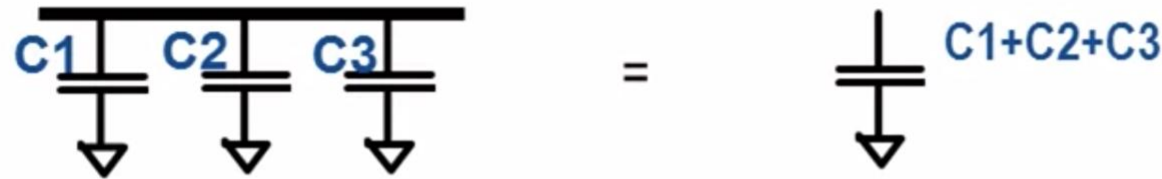
π Model

- Big idea: Replace **every** straight wire segment with **pi model**



RC Tree

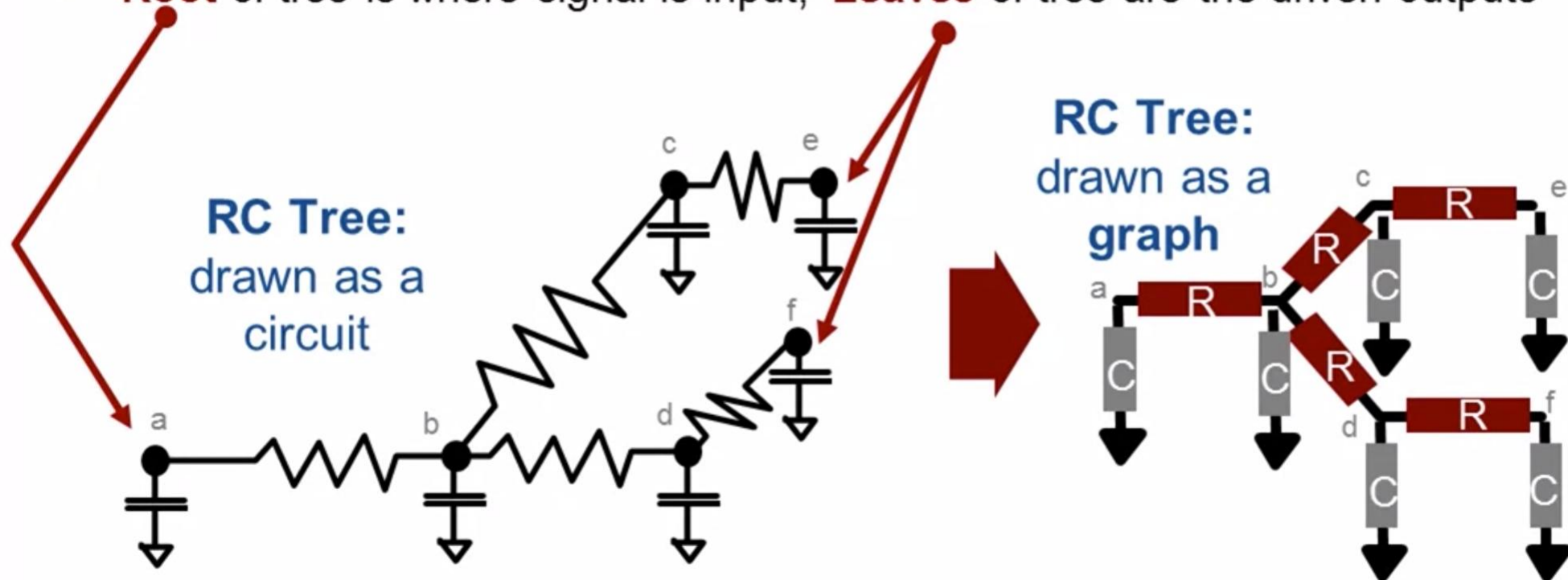
- **Simplification: Recall a rule from basic circuits (or physics)**
 - Parallel capacitors can be replaced by 1 capacitor with $\sum C_i$



Delay Graph

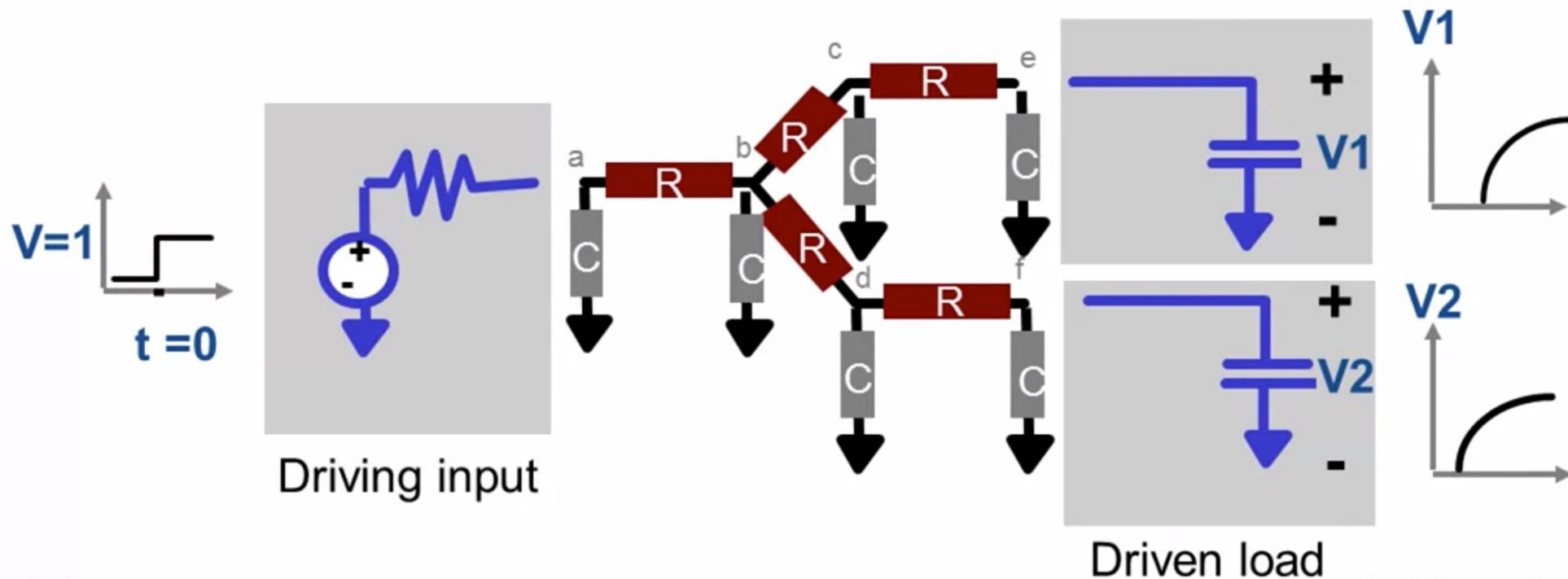
- **RC Tree general form**

- A **tree of resistors** (no loops); capacitors **“hanging off”** all intermediate tree nodes
- **Root** of tree is where signal is input; **Leaves** of tree are the driven outputs



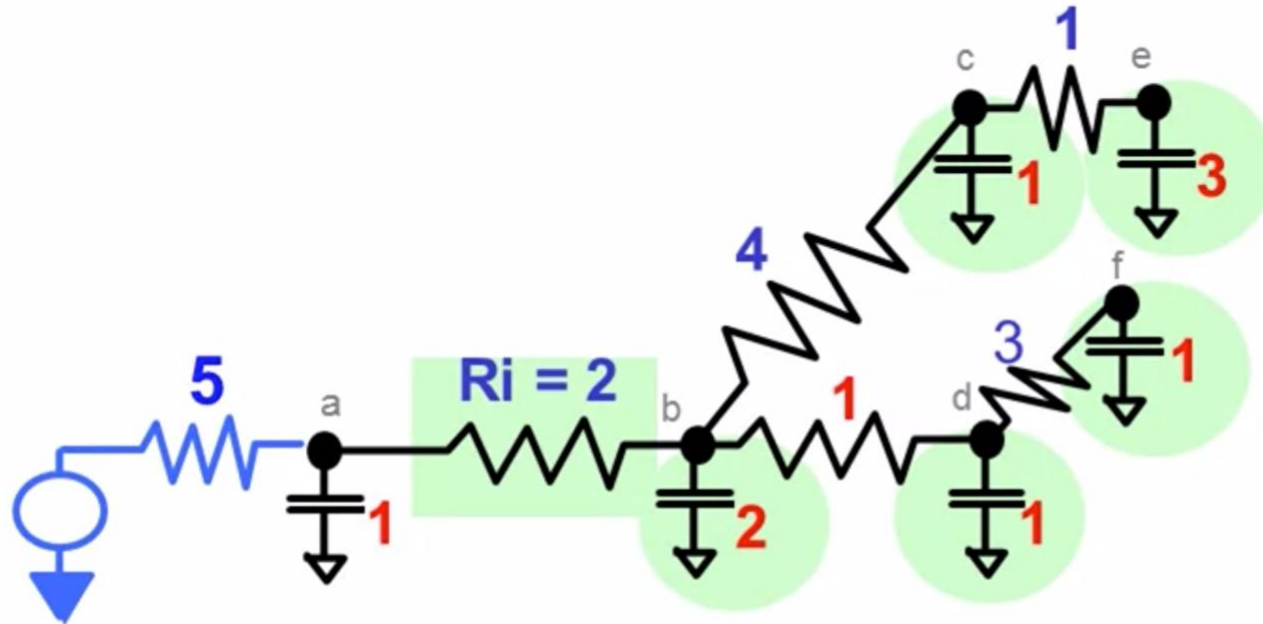
Driver & Load

- **Missing circuits detail: need to model driver, driven gates**
 - Voltage source + **resistor** as input at root (this models **driving** gate)
 - **Capacitor as load** at each leaf (each models a **driven** gate)



Elmore Delay

- **Do this:**
 - Set $\tau = 0$; walk down path of resistors from **Root** to **Leaf** where you want delay
 - At each resistor, do $\tau = \tau + R \cdot \sum (\text{all capacitors downstream})$
 - “**Downstream capacitor**” = any **C** that is reachable in tree below this resistor

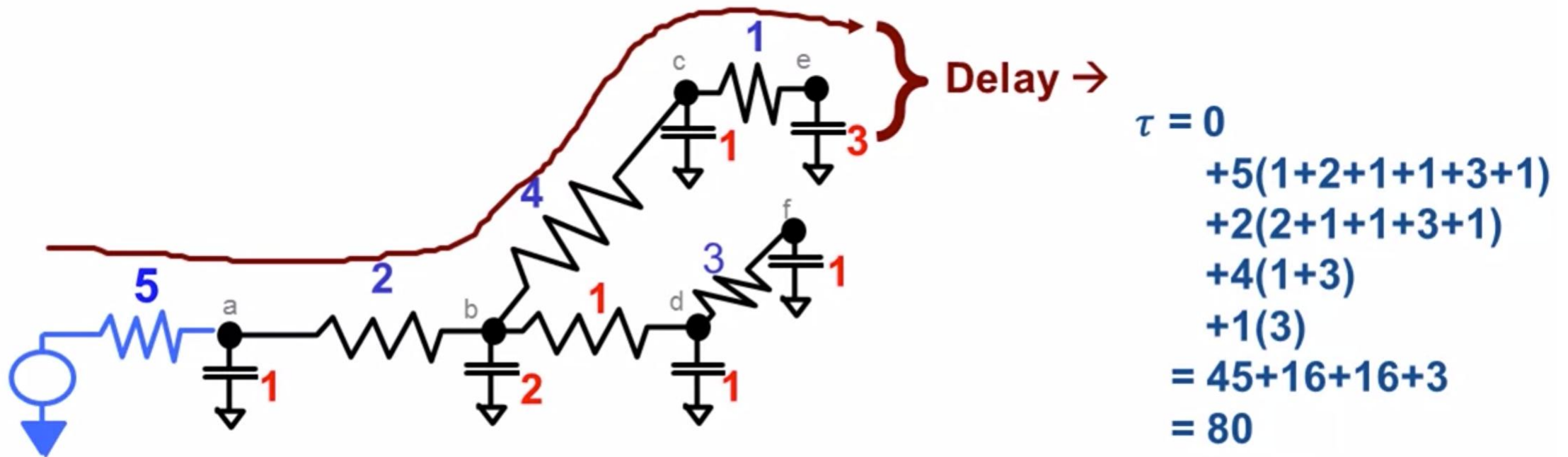


Example: at $R_i=2$ resistor in our RC tree, the term we would add to **Elmore delay** τ on a tree walk through R_i is $2 \times (2+1+1+1+3)$

Elmore Delay

- **Example:**

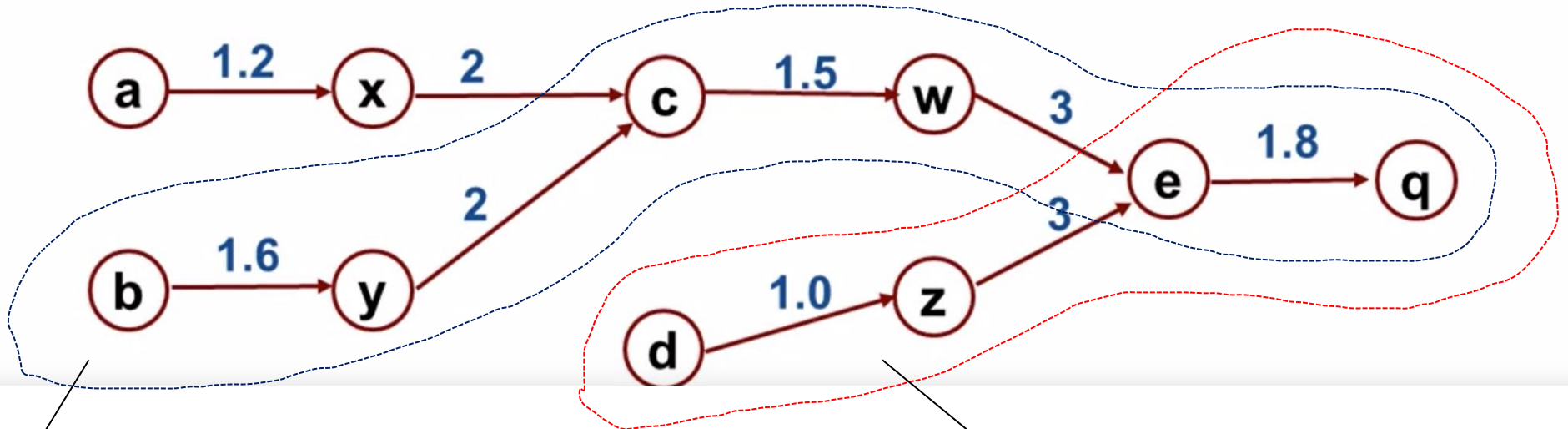
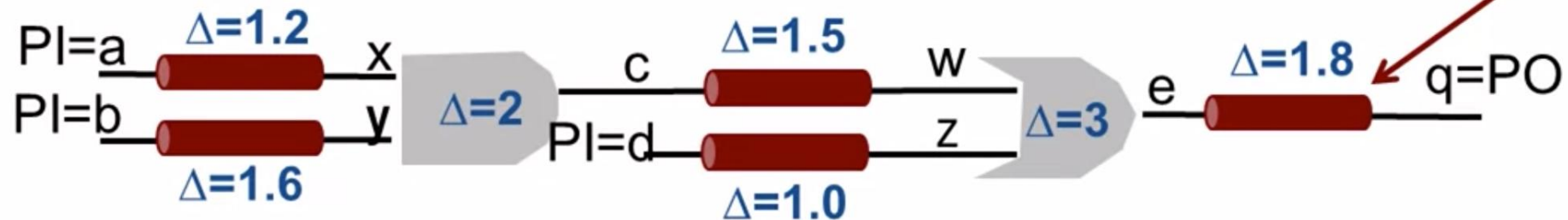
- Set $\tau = 0$; walk down path of resistors from **Root** to **Leaf** where you want delay
- At each resistor, do $\tau = \tau + R \cdot \sum (\text{all capacitors downstream})$



Path Delay

- What about **interconnect** delay?

- Can still use delay graph: model each wire as a “special” gate that just has a **delay**

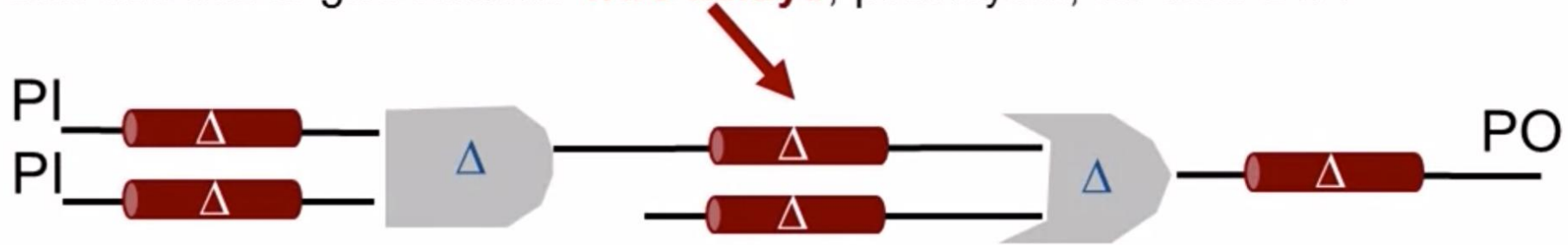


Max Path Delay = $1.6 + 2 + 1.5 + 3 + 1.8 = 9.9$

Min Path Delay = $1 + 3 + 1.8 = 5.8$

Delay Optimization

- Do people really use this delay metric? **Yes!**
- Timing verification
 - Can use this to give realistic **wire delays**, post layout, for final STA



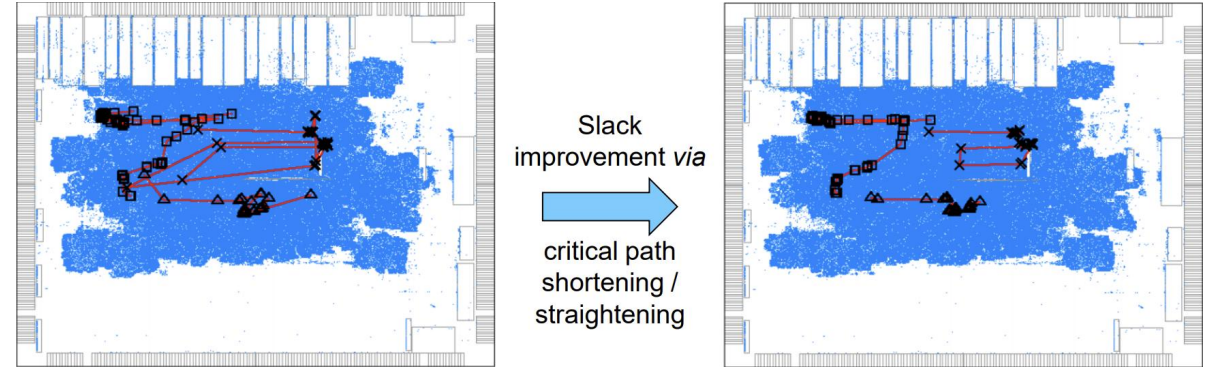
- During placement
 - Estimate **wire shape** (eg, a simple Steiner) you can get **very quick** delay estimate
 - Analytical placers use to **adjust weights** on wires, coerce critical wires to be **short**

Delay Optimization

- Interconnect has a huge impact on chip speed
 - Cannot ignore delays caused by the electrical properties of real wires
- Layout tools responsible for part of timing guarantee
 - Upstream tools determine levels of logic, gate count, fanouts, etc
 - Physical design tools responsible for how long the wires end up
 - All of these impact wire length and distribution
- Individual wires are today modeled as complex circuits
 - RC tree is the most useful model;
 - Elmore delay is easiest to compute
 - There are sophisticated estimators beyond Elmore
 - Can use for both verification, and for layout optimizations (eg clock)

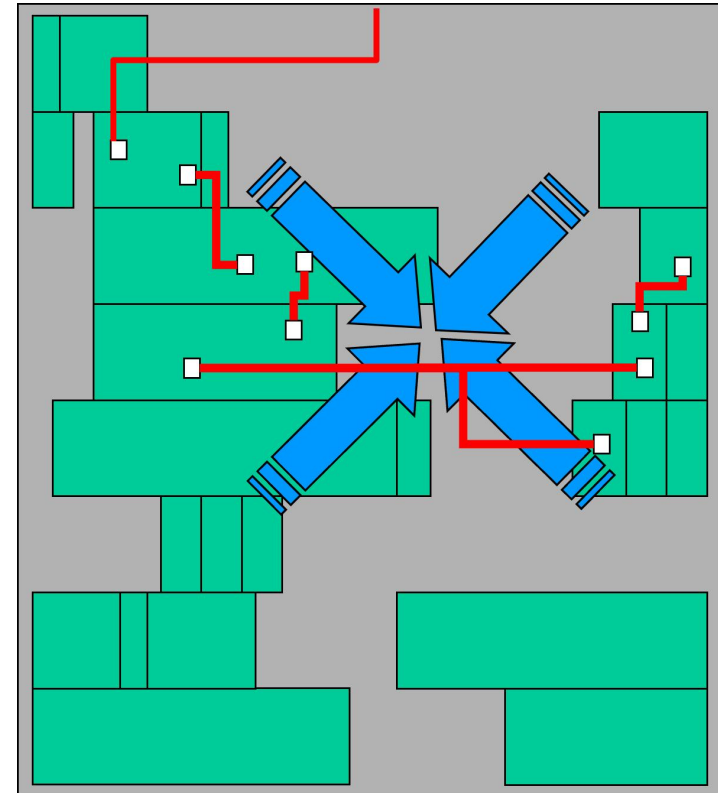
Timing-driven Placement

- Challenge
 - Cells **change greatly** in the **early stage**
 - May cause **new timing violations** to appear
 - Cells aggregate, **affecting the congestion**
- Problem description
 - **Total Negative Slack (TNS),**
 - **Worst Negative Slack (WNS)**
 - Get timing metrics from the timer to adjust cell coordinates
- Methodology
 - **Net-based approach:** Annotate the timing critical path information to the net level to **empower the critical net**, which can be easily adapted to all wire-length driving placements.
 - **Path-based approach:** According to the timing, a series constraint is added, and the position of the cells is updated by solving the **mathematical programming**.
 - **Differentiable methods:** Smoothing process of timing propagation process, and the timing metrics are added to the target of global placement.



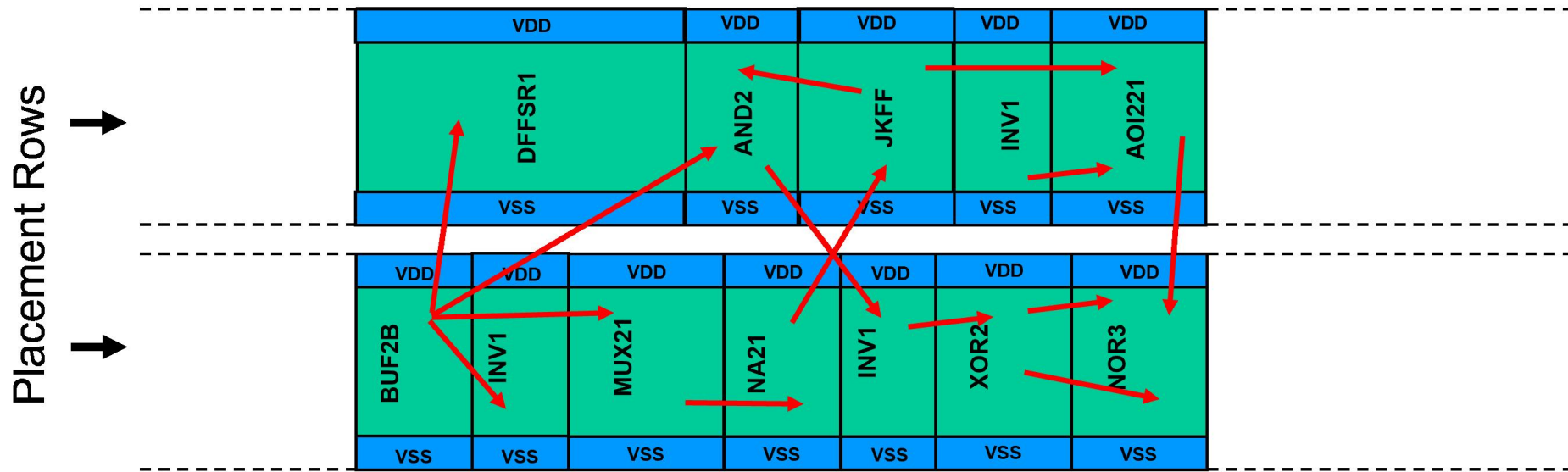
Timing-Driven Placement

- Timing-driven placement based on Virtual Route
 - Tries to place cells along timing-critical paths close together to reduce netRCs and meet setup timing
 - Net RCs are based on Virtual Routing (VR) estimates



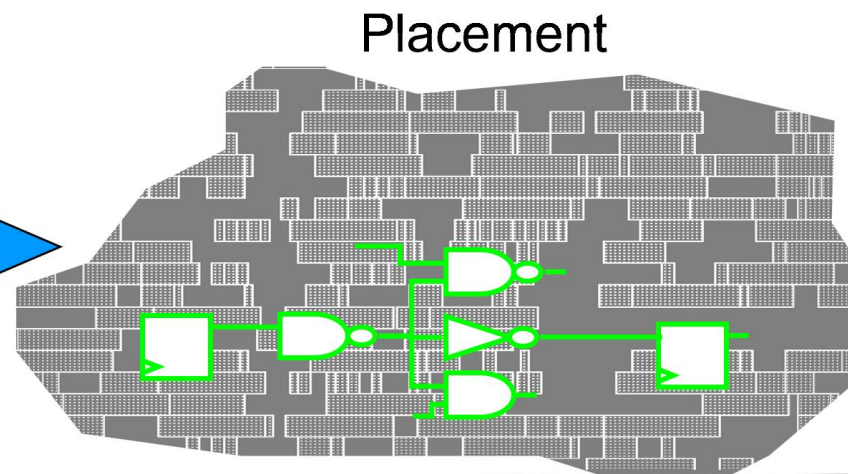
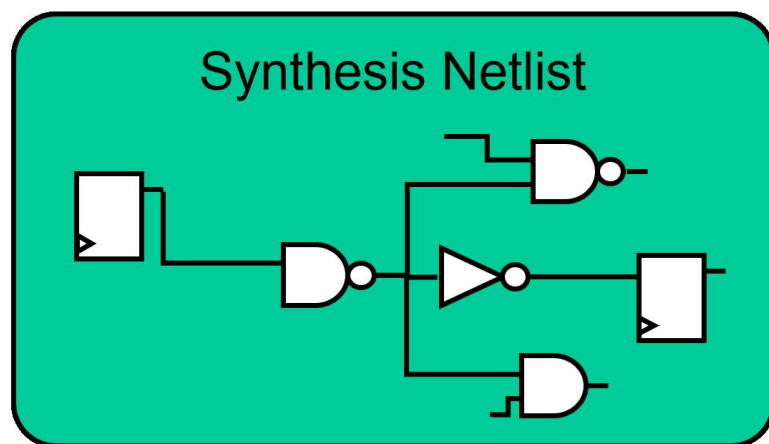
Timing-Driven Placement

- Standard cells are placed in “placement rows”
- Cells in a timing-critical path are placed close together to reduce routing-related delays → Timing-Driven Placement



Timing-critical cells placed together

TDP: Estimating Rnet and Cnet Before Placement



Net Fanout	Resistance K Ω	Capacitance pF
1	0.0498	0.045
2	0.1295	0.0812
3	0.2092	0.1312
4	0.2888	0.1811

Statistically Based Wire Load Model (WLM)

- 01 Introduction**
- 02 Placement Techniques**
- 03 Global Placement**
- 04 Detail Placement**
- 05 Timing Driven Placement**
- 06 Congestion Driven Placement**

Congestion

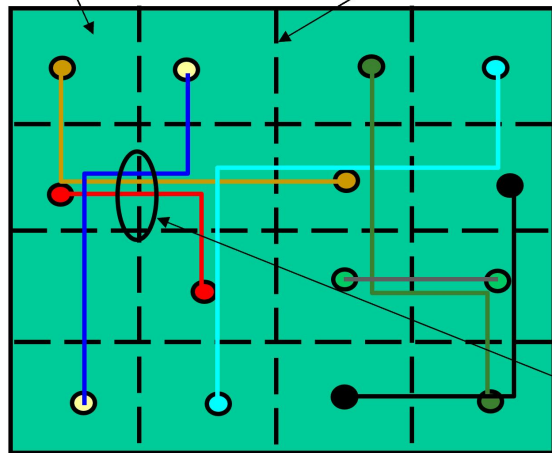
- Congestion occurs when the number of required routing tracks exceeds the number of available tracks.
 - Congestion can be estimated from the results of a quick global route.
 - Global bins with routing

$$overflow_{i,j} = \max [0, (\text{demand}_{i,j} - \text{capacity}_{i,j})]$$

$$util_{i,j} = \text{demand}_{i,j} / \text{capacity}_{i,j}$$

$$\text{capacity}_{i,j} = \left\lfloor \frac{\text{grid_h}}{d_{\text{pitch}}(\text{layer})} \right\rfloor$$

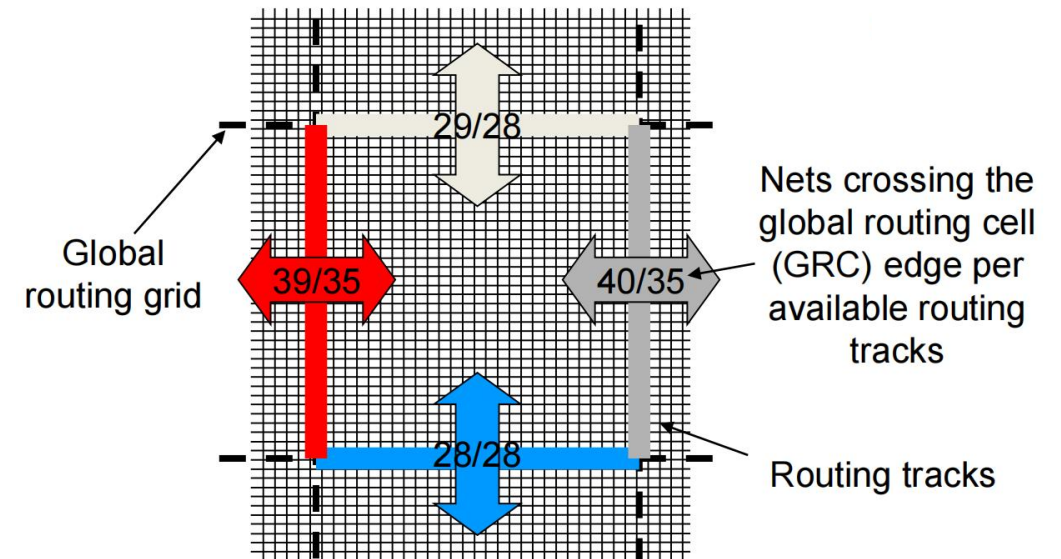
Global Bin Global Bin Edge



Overflow on each edge =
 $\begin{cases} \text{Routing Demand} - \text{Routing Supply} \\ 0 \text{ (otherwise)} \end{cases}$

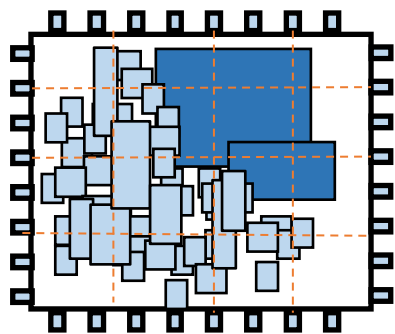
Total Overflow = $\sum \text{overflow}$
all edges

Routing demand = 3
 Assume routing supply is 1,
 overflow = 3 - 1 = 2 .

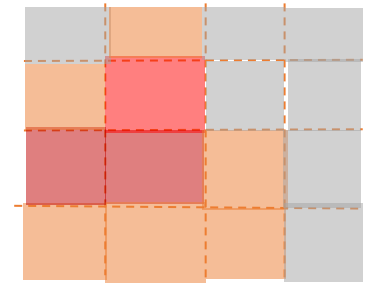
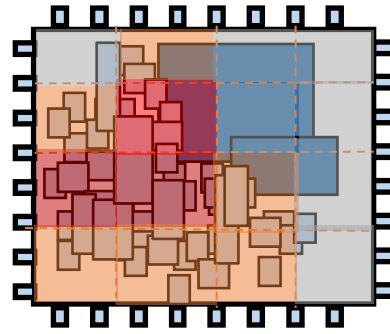


Congestion

- 设计的规模在不断增大然而METAL的层数却是固定的，导致布线的难度上升，对布线的power和routability提出了新的挑战。可布线性的问题可以在布局阶段进行考虑。该问题的目标是通过嵌入拥塞估计的工具（例如全局布局工具）得出拥塞评估进行布局。



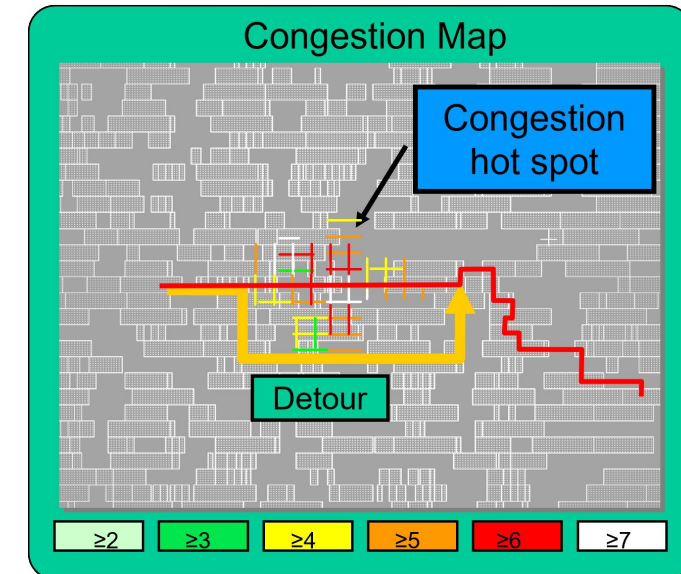
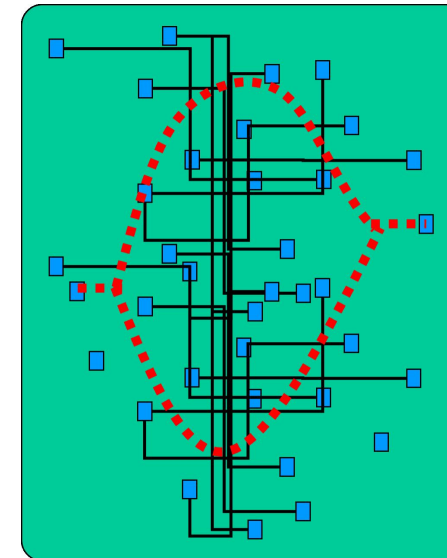
Global routing



Congestion map

Placement Issues with Congestion

- Issues with Congestion
 - If congestion is not too severe, the actual route can be detoured around the congested area
 - The detoured nets will have worse RC delay compared to the VR estimates
 - In highly congested areas, delay estimates during placement will be optimistic.
- Not routable or severely congested design
 - It is important to minimize or eliminate congestion before continuing
 - Severe congestion can cause a design to be un-routable



Congestion Evaluation

- Existing Methods:

- Static methods: **Cell Density, Pin Density**

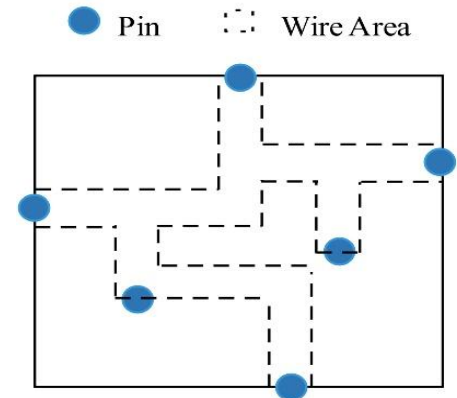
- provide limited information, like pin density, and are fast but less accurate in representing congestion.

- Constructive methods: **Run Global Routing**

- offering high accuracy but can be slow due to router limitations.

- Probabilistic methods: **RUDY**

- analyze wire density distribution, balances evaluation speed and accuracy but makes unrealistic assumptions that limit precision.



(RUDY Conceptual Figure)

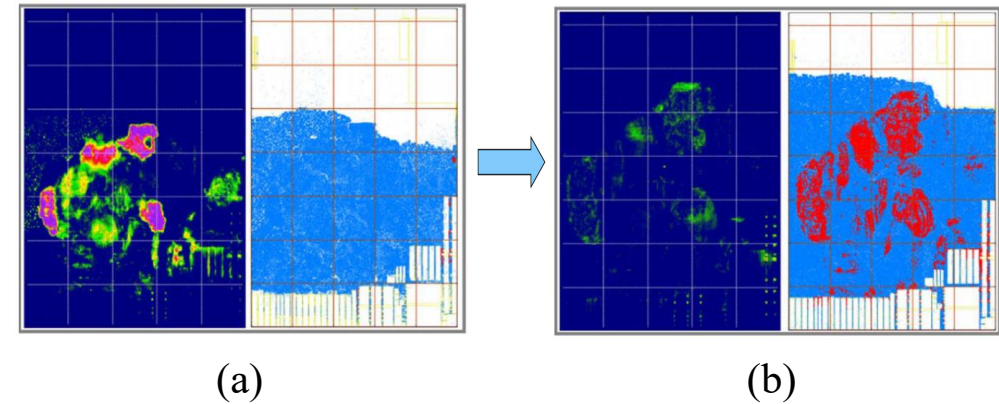
$$\frac{D(e)}{C(e)} = \sum_{i=1}^m \frac{WL(i) \times Width(i) \times Overlap(i)}{BBoxArea(i) \times C(e)}$$

- m is the number of nets,
- $\frac{D(e)}{C(e)}$ is the congestion value of each GRC e ,
- $c(e)$ is the capacity of GRC e ,
- $Overlap(i)$ is the overlap area ratio between net_i and the GRC e .

Routability-driven Placement

- Challenge

- Cells **change greatly** in the **early stage** of placement
- Difficult to **balance** performance and evaluation
- Increasing wire length and timing **damage** issues



- Problem description

- Common evaluation metric: **Total Overflow (TOF)** , **Maximum Overflow (MOF)**
- Routability-driven placement adjusts the cell position **according to the congestion map**

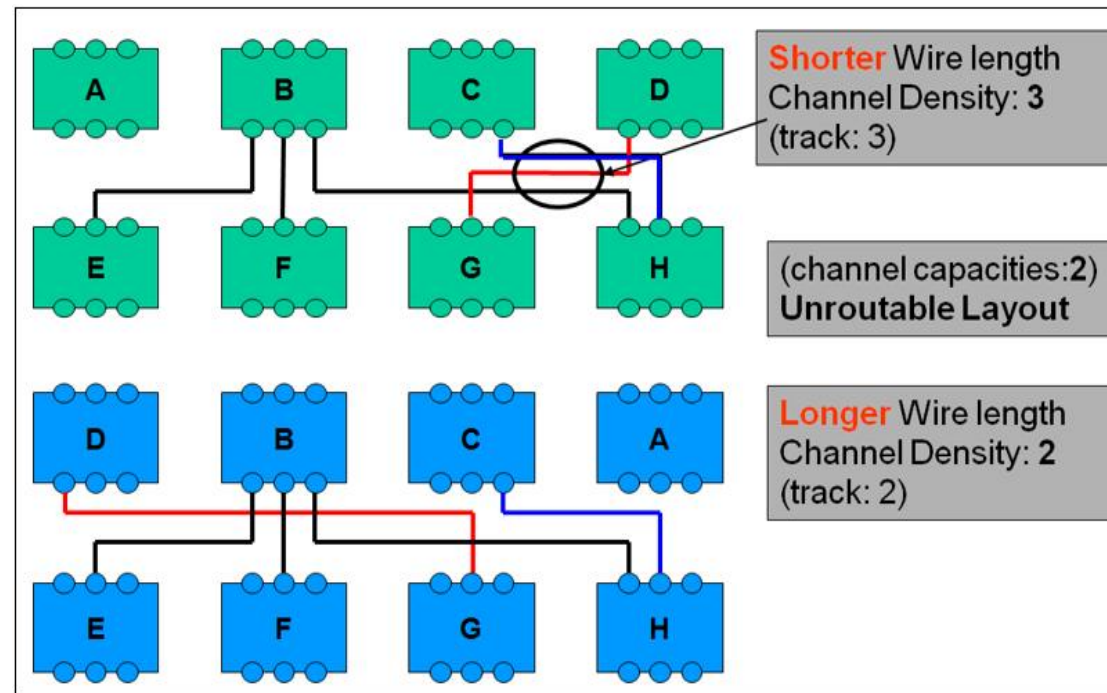
- Methodology

- Evaluation: Static, probabilistic, constructed, and network methods. At present, the **commonly used methods** are the **probability** method and the **construction** method
- Optimization: Adjust the density to **indirectly optimize** congestion, and directly integrate congestion items to **participate in optimization**

Congestion-driven Placement

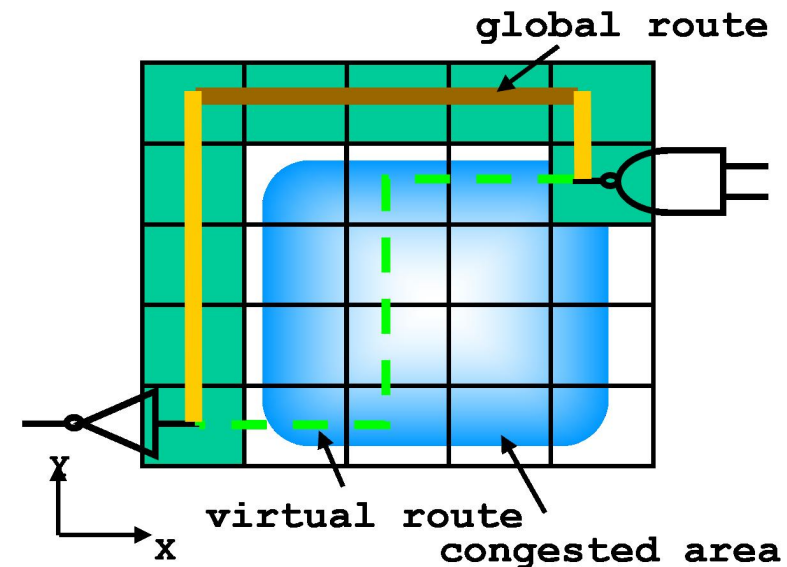
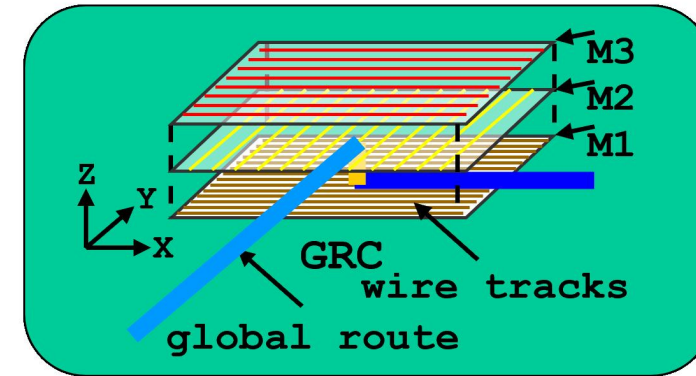
- Congestion Reduction

- The tool tries to evaluate congestion hotspots and spread the cells (lower utilization) in the area to reduce congestion.
- The tool can also choose cell location based on congestion, rather than wire-length.

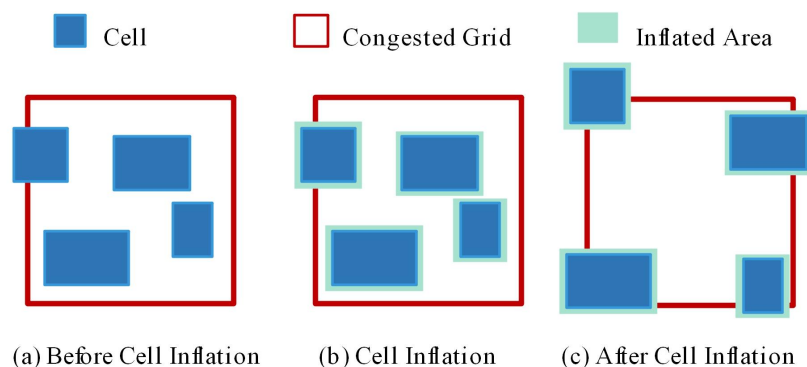


Global Route (GR) for Congestion Map *iEDA*

- GR assigns nets to specific metal layers and global routing cells (GRCs)
 - The detailed router will follow GRCs path
- GR determines if each GRC along a path has enough wire tracks for assigned nets
- If not enough wire tracks, GR reassigns metal layers or GRC accordingly



- Indirect Congestion Optimization via Density Tuning
 - Cell Inflation:
 - Temporarily enlarge congested grid cells.
 - Key considerations: which cells to inflate, direction, and amount.
 - Threshold Density Adjustment:
 - Decrease the density threshold of congested grid cells, such as based on pin density corrections.
- Direct Congestion Representation in Objective Function
 - Smooth Net Congestion:
 - $\min \lambda_1 WL + \lambda_2 \sum (D - M_b)^2 + \lambda_3 \sum (C - S_b)^2$
 - Smooth Pin Density:
 - $\min \lambda_1 WL + \lambda_2 \sum (D - M_b)^2 + \lambda_3 P$
 - Congestion Weighting on Wirelength:
 - $WL = \sum (\alpha_e \max |x_i - x_j| + \beta_e \max |y_i - y_j|)$
 - $\alpha_e = 1 + (f_y - f_x), \beta_e = 1 - (f_y - f_x)$



(The process of cell inflation)

(Comparison of Strategies for Cell Inflation Methods)

Tool	Time	Cell Selection	Inflation Direction	Inflation Rate
SimPLR	2011	Top 5% Most Congested Grids	Horizontal	Simple Weighting Function
Ripple	2013	All Congested Grids	Horizontal/Vertical	Simple Piecewise Function
POLAR	2014	Top 10% Most Congested Grids	Unspecified	Fixed at 10%
RePIAce	2019	All Congested Grids	Unspecified	Superlinear Function

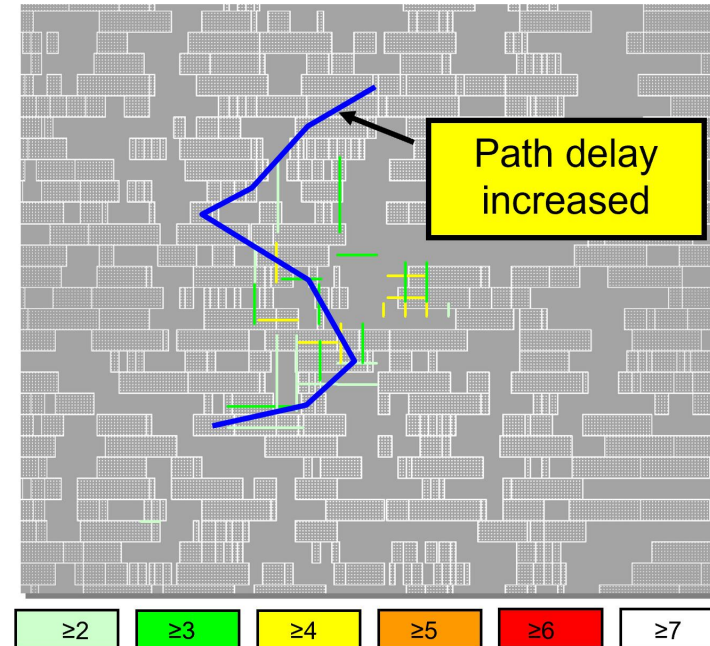
Strategies to Fix Congestion

- Modify the floorplan:
- Mark areas for low utilization.
- Top-level ports
 - Changing to a different metal layer
 - Spreading them out, re-ordering or moving to other sides
- Macro location or orientation
 - Alignment of bus signal pins
 - Increase of spacing between macros
 - Add blockages and halos
- Core aspect ratio and size
 - Making block taller to add more horizontal routing resources
 - Increase of the block size to reduce overall congestion
- Power grid
 - Fixing any routed or non-preferred layers

Congestion vs. Timing-Driven Placement *iEDA*

CONGESTION ← → **TIMING**

- Cells along timing critical paths can be spread apart to reduce congestion
- These paths may now violate timing



Small timing violations can be resolved later by incremental logic optimization.

Conclusions

- Placement
- Placement Techniques
 - Wirelength Estimation
 - SA
 - Quadratic Placement
- Global Placement
 - Wirelength Smoothing, Density Smoothing, Gradient Optimization
- Detail Placement
 - Legalization, Detail Placement
- Timing Driven Placement
- Congestion Driven Placement



最新动态



开源EDA RISC-V

2024-8-20

iEDA团队在第四届RISC-V中国峰会组织
OSED论坛



EDA 芯片 开源 智能

2024-7-20

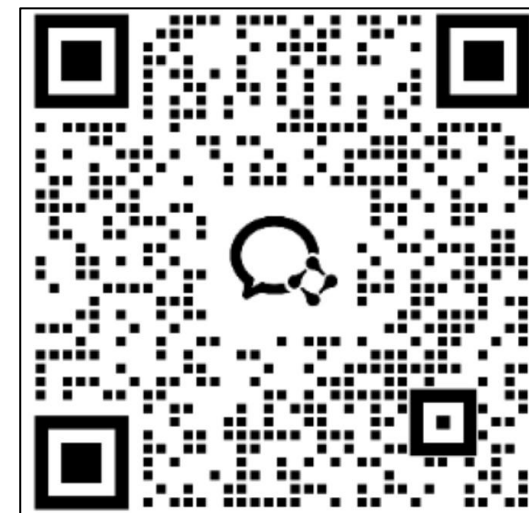
iEDA团队在第二届CCF芯片大会组织开源
智能EDA论坛



EDA DAC

2024-06-24

iEDA团队参加61届Design Automation



Thanks

iEDA website: ieda.oscc.cc

李兴权 (Xingquan Li)
fzulxq@gmail.com