Physical Design

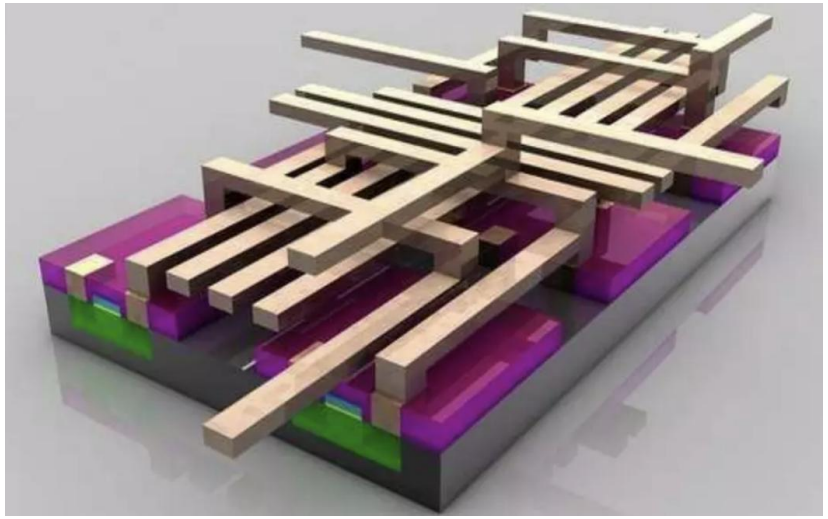
Routing

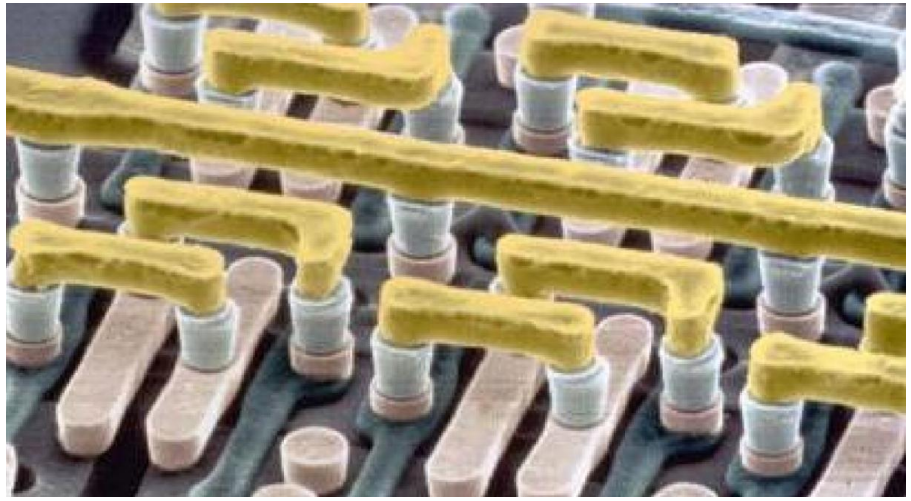
李兴权

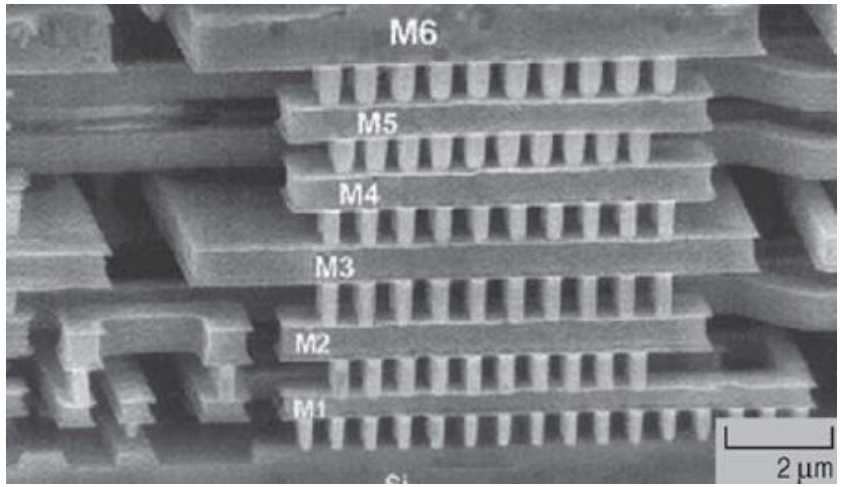
iEDA

iEDA

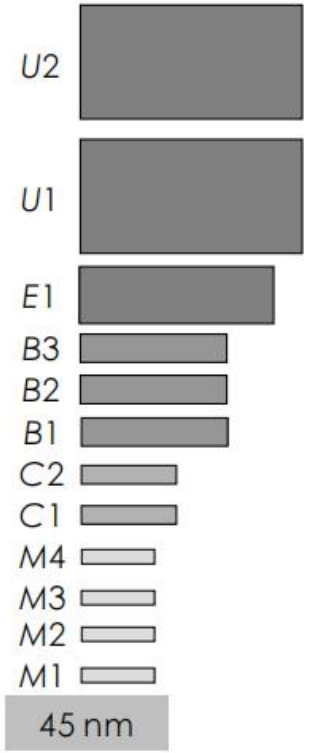
布线





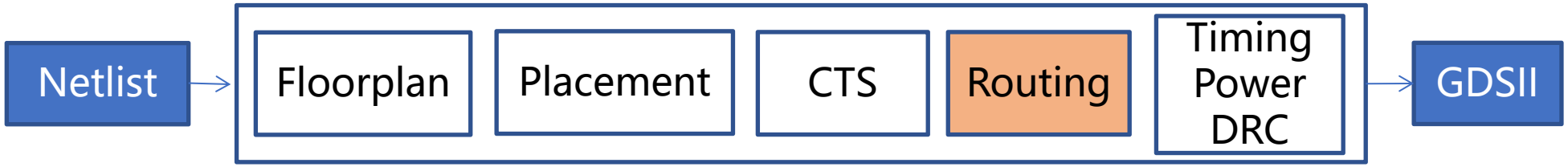


UMC 6 metal stack

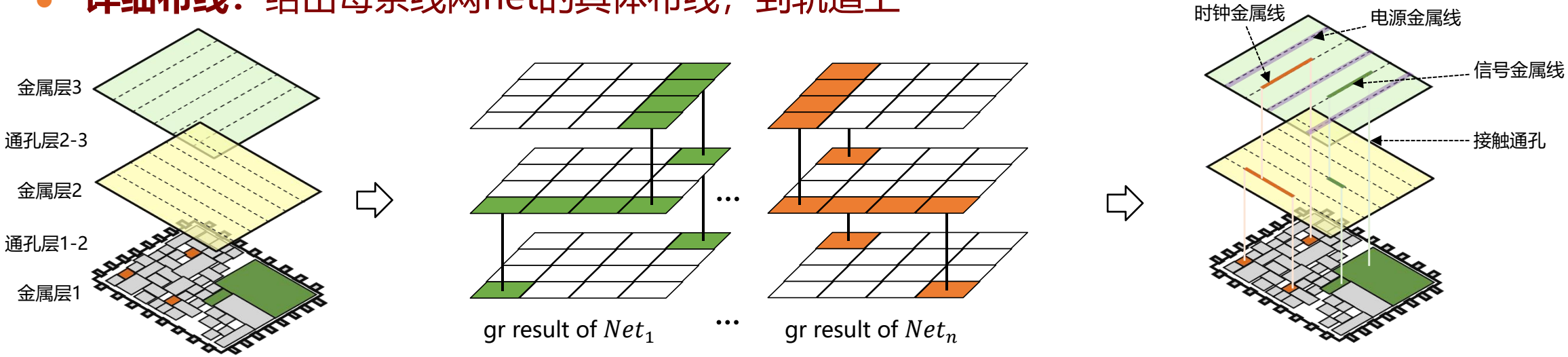


布线步骤

- Routing in physical design

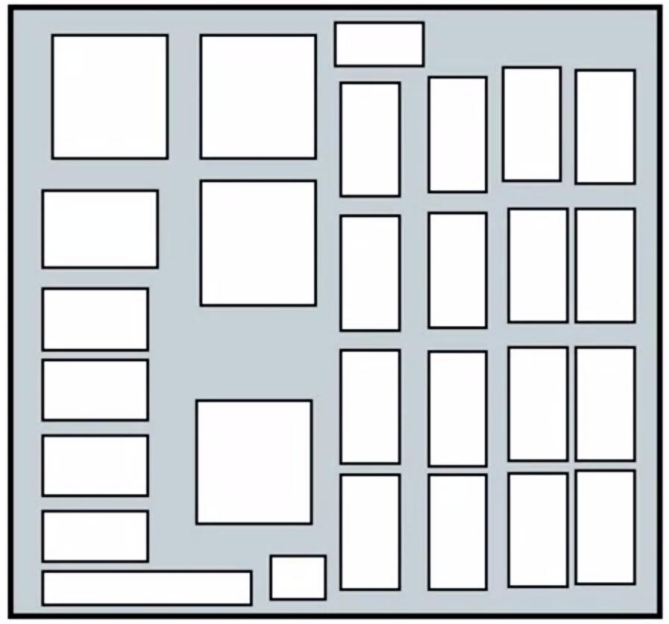


- **全局布线：**给出每条线网net宽的、大致的布线通道，指导详细布线的进行
- **详细布线：**给出每条线网net的具体布线，到轨道上

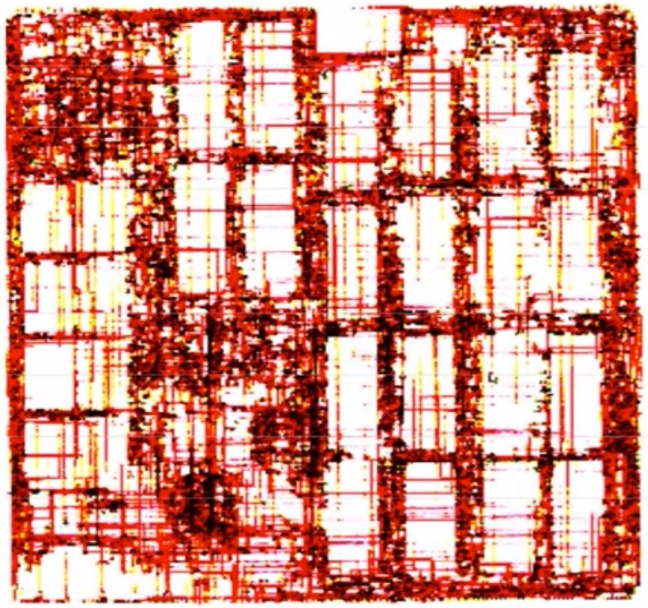


全局布线　　**详细布线**

布线挑战



Thousands of macro blocks
Millions of gates
Millions of wires

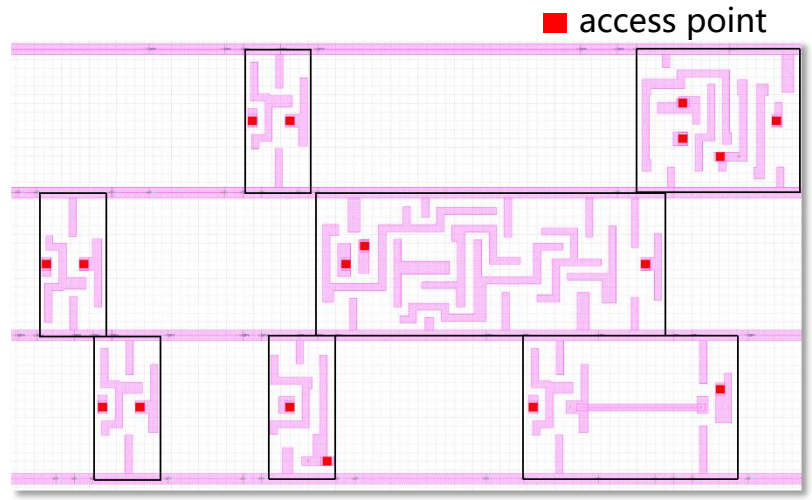


Kilometers of wire.

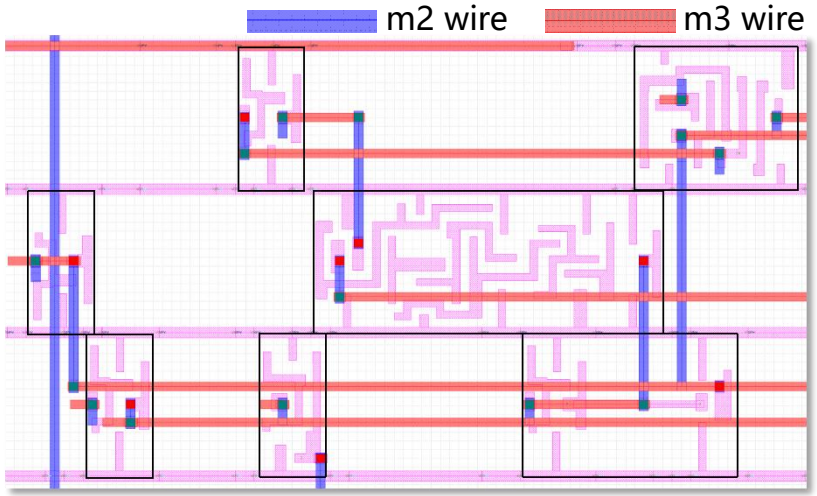
- Big chip, 1cm x 1cm
- 20-50 million nets
- Modern IC technology, ~100nm pitch for wires (grid size)
- So, 100K x 100K routing grid!
- x10 routing layers?
- 100 billion grid cells!
- Do we really do it like this?

布线输入输出

- 问题描述：布局完成后，每个Net的连接的Pins的位置都相对固定。布线是通过建立合法的路径将每个Net上的Pins在物理上实现**连通，无短路**。同时满足设计制造规则和时序约束。
- 输入：
  - lef (物理形状,布线资源)
  - def (Pin坐标,线网关系)
  - lib (器件时序信息)
  - drc (物理设计规则)
  - sdc (时序约束)
- 输出：
  - gds(带有Wire，Via，Pin，Cell等信息)
- 目标：
  - 线长，时序、功耗...
- 约束：
  - 在线网连通无短路的情况下，满足设计制造规则，时序规则...



Top view before routing



Top view after routing

Routing大纲

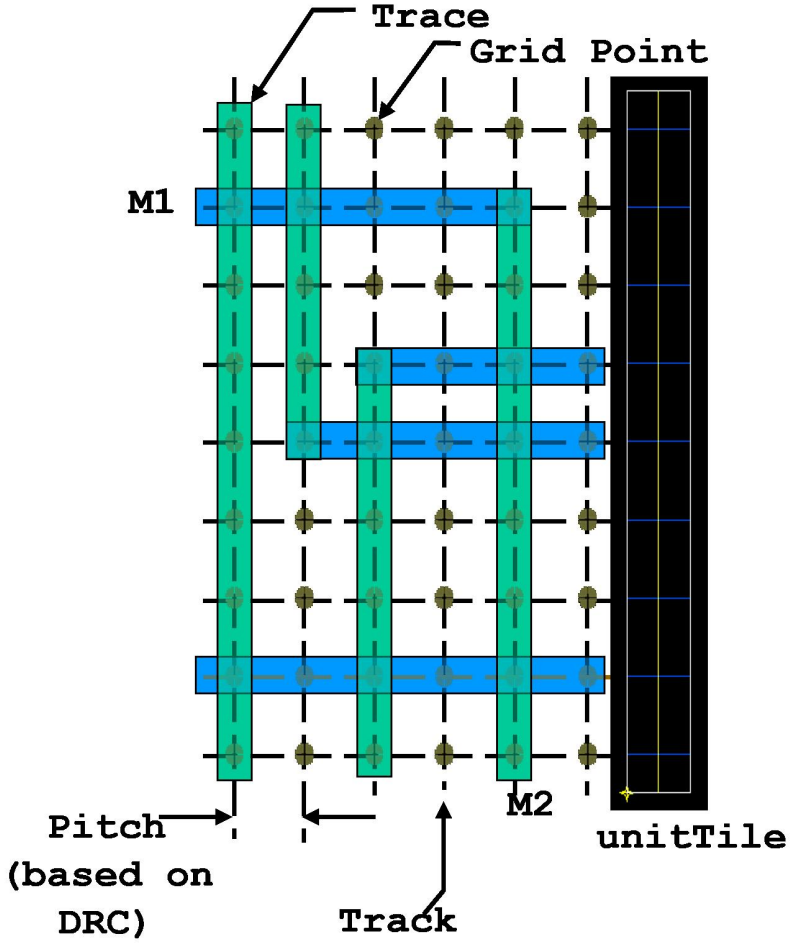
iEDA

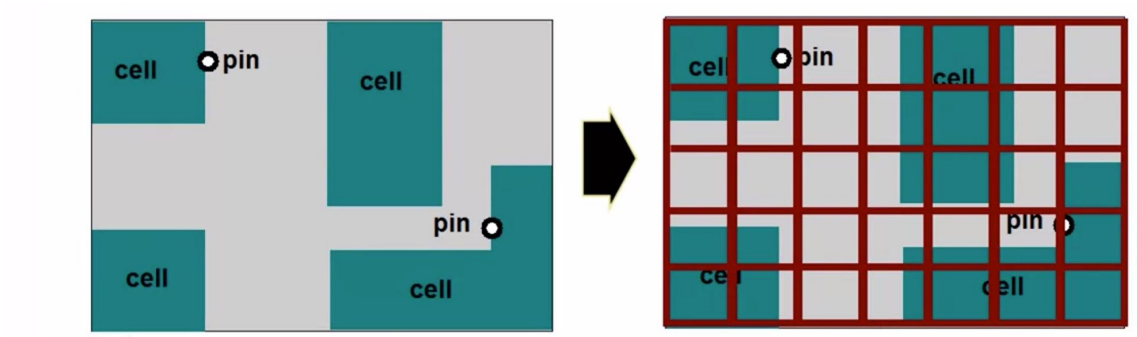
01—资源分配
0101—离散化数据结构选取
0101S01—GCell
0101S02—SlotPolygon
0102—全局均匀化求解
0102S01—激活集法
0102S02—罚方法
0102S03—多级框架下均匀化求解
0103—避障下的均匀化求解
0103S01—罚方法下的伪避障求解
0104—无隔断均匀化求解
0105—单元可移动的均匀化求解
0106—Congestion估计
0106S01—FastRoute
0106S02—BBox初值重叠
0106S03—net虚连线估计
0107—机器学习下的Congestion估计

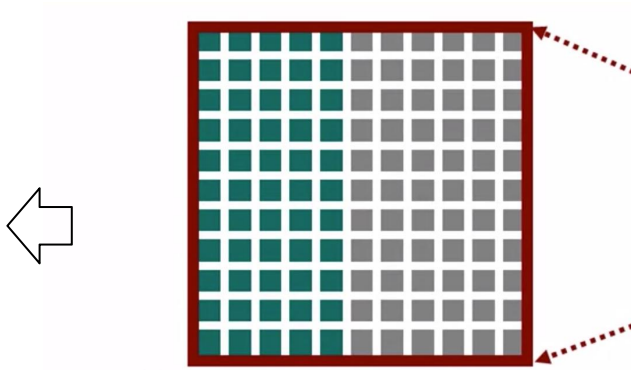
02—全局布线
0201—直角斯坦纳树生成
0201S01—Flute(建表)
0201S01—Flute(查表)
0202—基于驱动负载关系的斯坦纳树
0203—基于避障的斯坦纳树
0204—基于边值赋权的斯坦纳树
0205—单元可移动的斯坦纳树
0206—斯坦纳树拓扑相似性学习
0207—两点布线
0207S01—maze布线
0207S02—A*
0208—加权图两点布线
0208S01—权值A*
0209—强化学习下的两点布线
0210—线网布线
0211—层分配

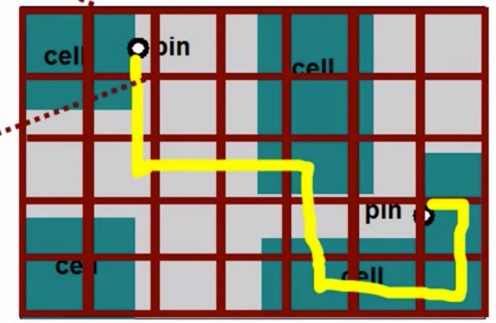
iEDA

网格化布线系统





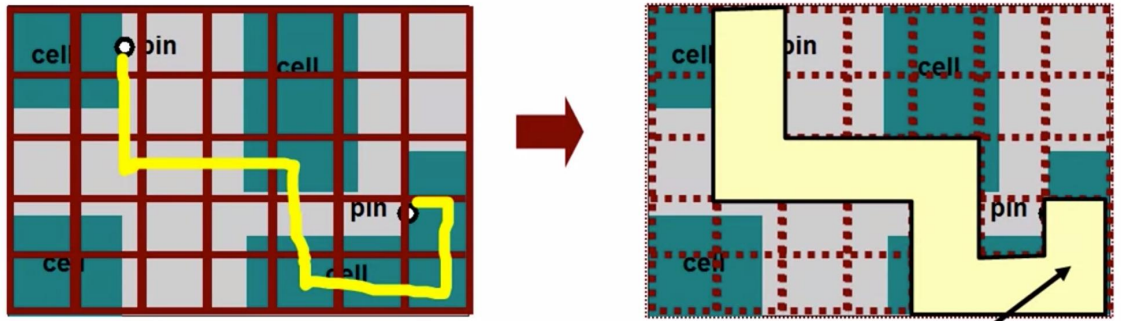




All the routing we have done so far is called **Detailed Routing**.
Means we plan exact, final path of wire.
Each cell in **Global Routing** grid is a box (called a **GBOX**) with size typically 100-200 wire grids on a side.
(This example just has ~10 for clarity)

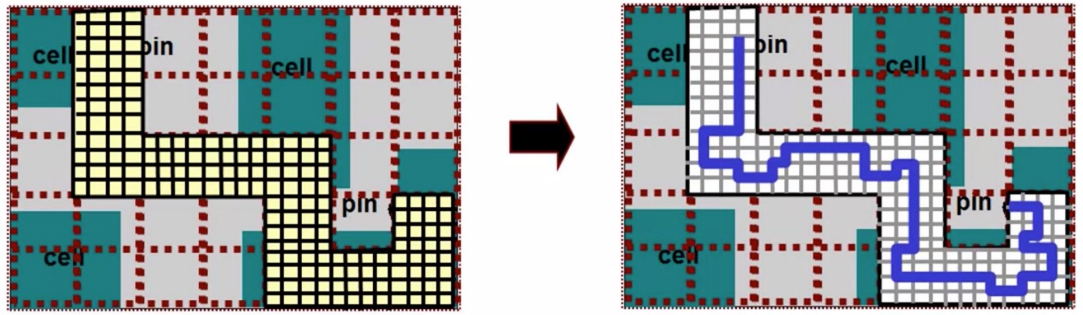
再看全局布线和详细布线

- **Makes sure overall wiring congestion is reasonable**
  - Balance **supply** (how much available space) vs **demand** (how many paths *want* to go here)
  - Global routing generates **regions of confinement** (ie, coarse path) for a wire



全局布线

- **Detailed routing embeds *exact* paths in these regions**
  - Simplest model is **grid**: require wires, pins to use tracks **on** this grid



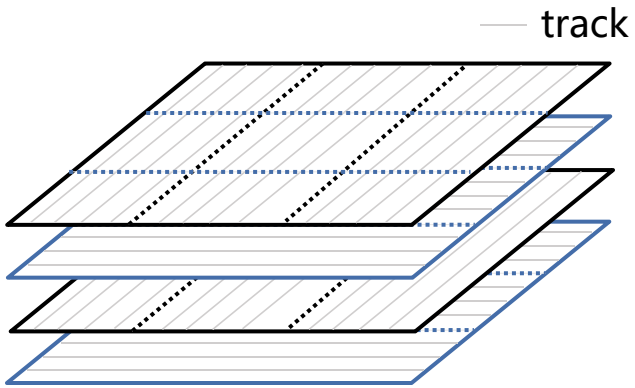
Global router tells us to search for detailed paths only here

Detailed router tells us exact final path in this region

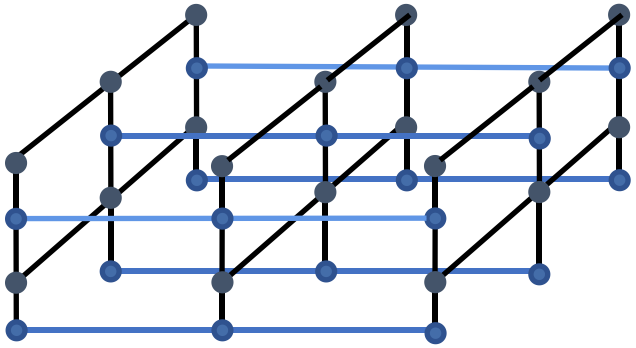
详细布线

GCell

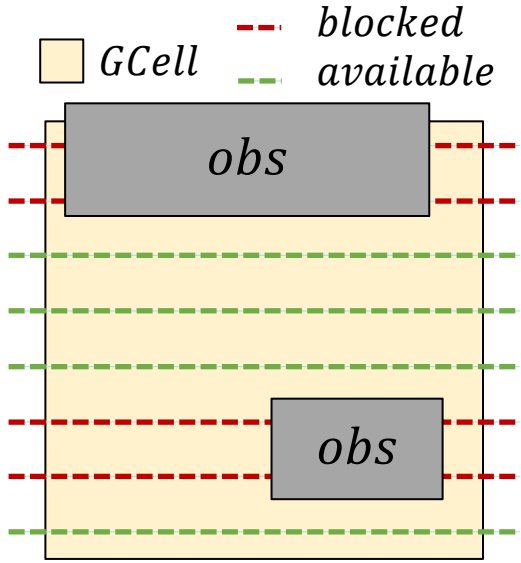
- **GRG（Global Routing Grid），GCell (Global Cell)**



The layout is multi-layered, with each layer consisting of uniform routing tracks. Each layer has a corresponding preferred direction.



Using GCell to partition the layout and construct a three-dimensional graph.

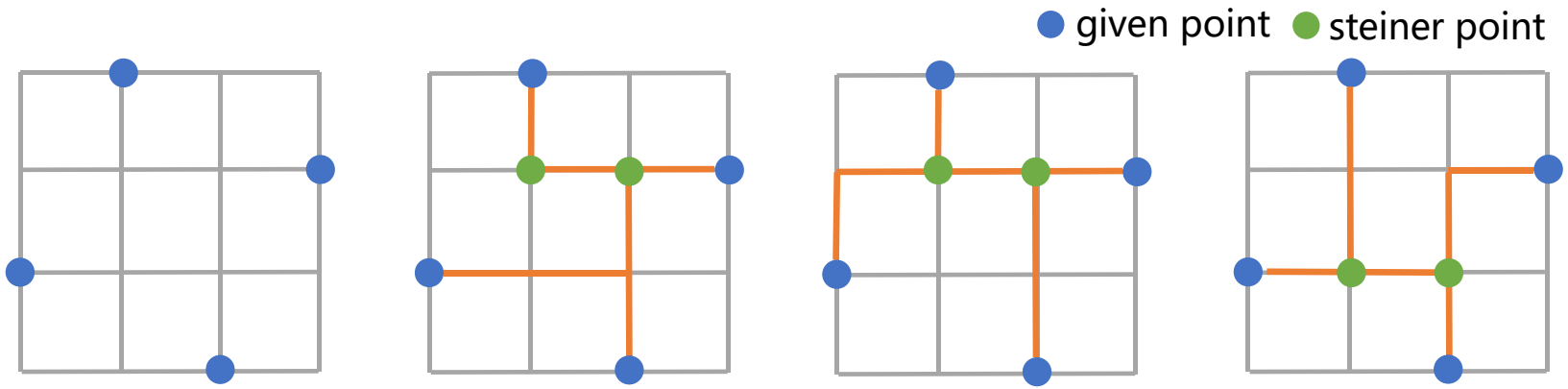


The routing resources within the GCell are as shown in the diagram, where the red portion represents obstructed resources, and the green portion represents available resources

Rectilinear Steiner tree (RST)

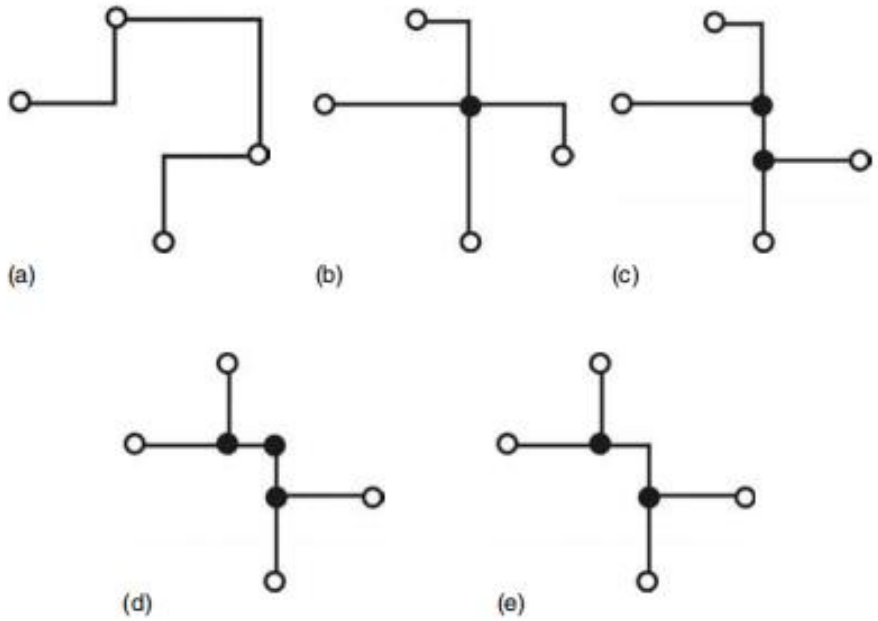
iEDA

- Rectilinear Steiner tree (RST)
  - Given a set of points
  - Additional points (Steiner points) can be introduced
  - Limited to horizontal and vertical segments only



迭代1-斯坦纳算法

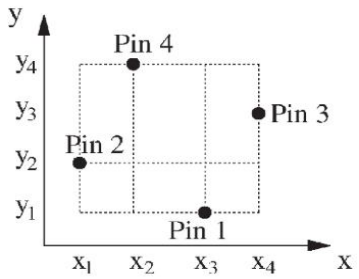
- 迭代1-斯坦纳算法



Flute(建表)

- 描述：Flute通过离线建表，在线查表的方式，可以快速得到一个net的若干个斯坦纳树，本页描述了如何建表。

1



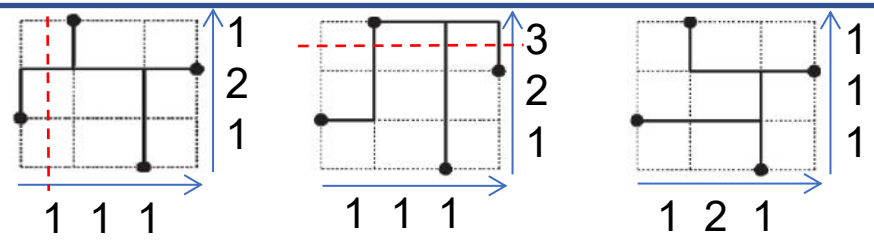
(3 1 4 2)

pin的index排序依据y坐标index升序

$Pin1(x3, y1)$ $Pin2(x1, y2)$ $Pin3(x4, y3)$ $Pin4(x2, y4)$

再以x坐标的index，即得到一个序列3142

2

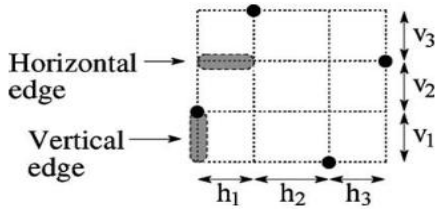


POWVs (1 1 1 1 2 1) (1 1 1 1 2 3) (1 2 1 1 1 1) ....

$(a_1a_2a_3a_4a_5a_6)$

通过遍历Hanan grid生成此net若干个可能的斯坦纳树，并且生成对应的POWVs(potentially optimal wirelength vectors)，如左图红虚线穿过黑线的数量，POWVs记录的顺序为先行再列(蓝箭头指向顺序)。

3



通过计算每个POWVs的WL比较筛选，最后得到最优值。

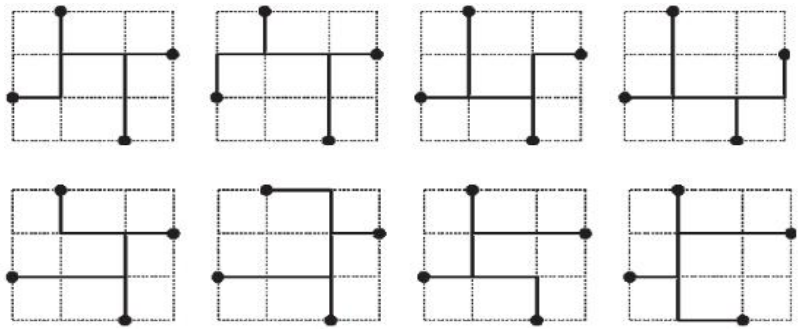
$$WL = h_1 * a1 + h_2 * a2 + h_3 * a3 + v_1 * a4 + v_2 * a5 + v_3 * a6$$

POWVs

(1 1 1 1 2 1)

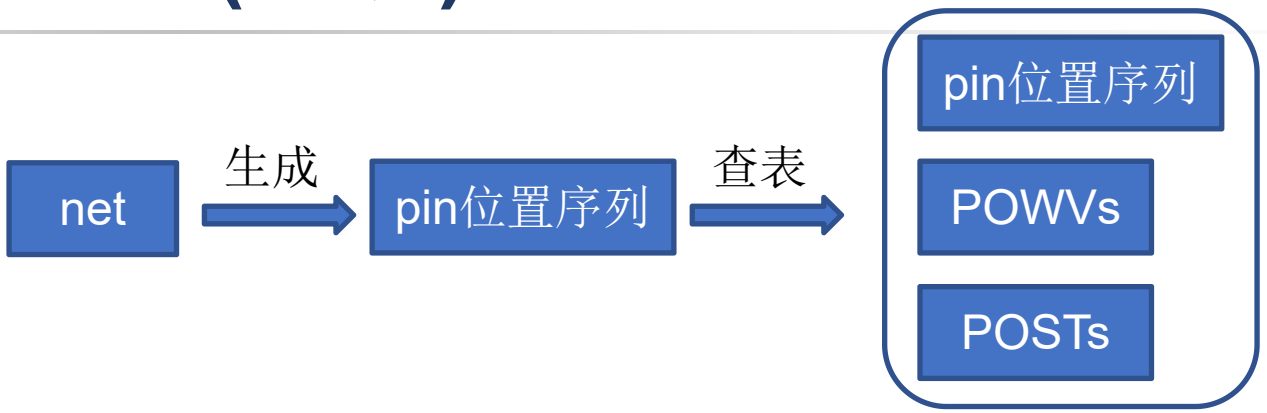
(1 2 1 1 1 1)

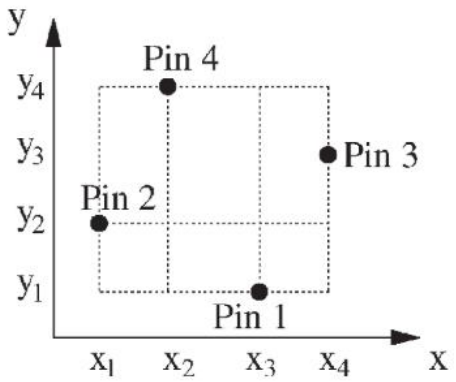
POSTs(potentially optimal Steiner trees)



Flute(查表)

- 描述：Flute通过离线建表，在线查表的方式，可以快速得到一个net的若干个斯坦纳树，本slide描述了如何查表。





pin位置序列

3142

POWVs

.....

1 2 1 1 1 1

1 1 1 1 2 1

.....

POSTs

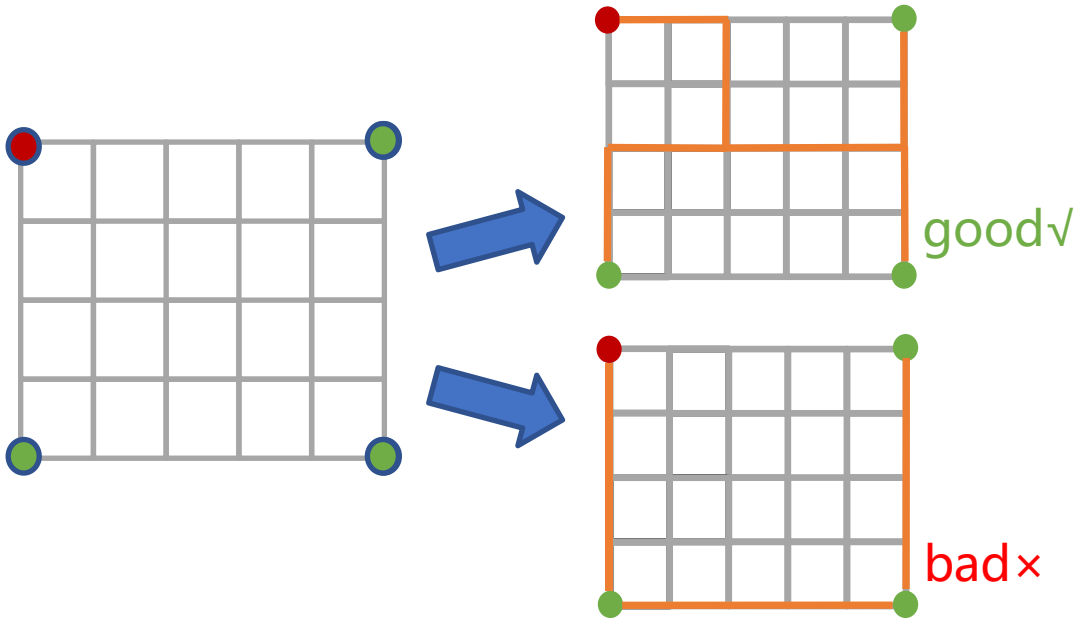
.....

.....

驱动负载平衡的斯坦纳树

iEDA

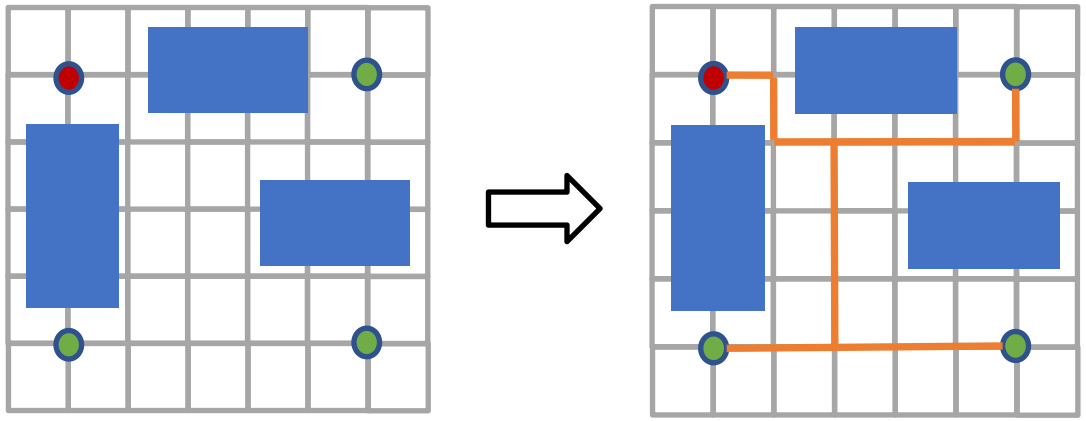
- 问题描述：若一个net有k个pin脚，那么必有一个输出pin脚(驱动)，和k-1个输入pin脚(负载)，在生成net对应的直角斯坦纳树(RSMT)时，希望在生成路径足够短的情况下，驱动到各个负载之间的距离之差尽量小。在建立斯坦纳树时，斯坦纳点往往出现在Hanan网格上，可以通过Hanan网格大幅度缩短解空间。
- 目标：
  - 连线最短
  - 驱动到各个负载的曼哈顿距离之差最小



避障斯坦纳树

iEDA

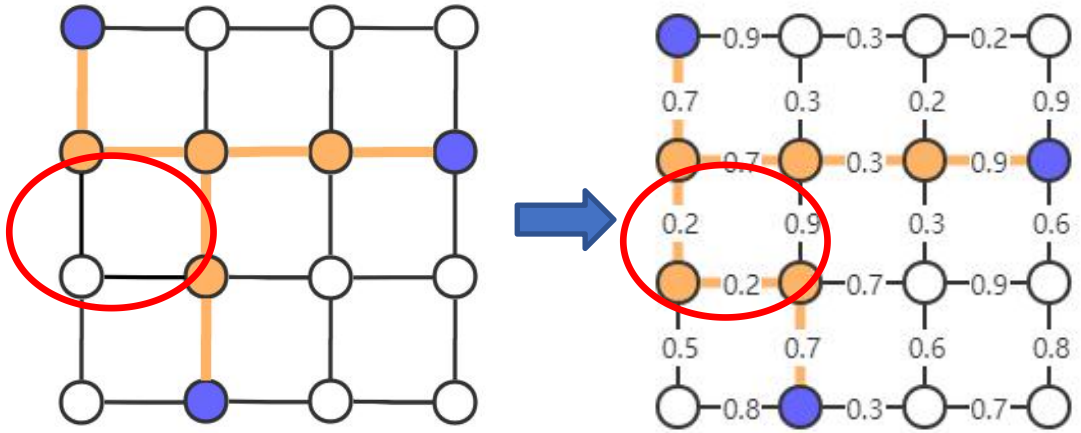
- 问题描述：给定若干个点和一些障碍，需要在绕过障碍的同时，将这些点连起来，即在避障时生成对应的直角斯坦纳树(RSMT)，在建立斯坦纳树时，斯坦纳点往往出现在Hanan点上，可以通过Hanan点大幅度缩短解空间，障碍的存在也会使得Hanan点减少，但更糟糕的是，可能不存在任何一个Hanan点。



边值赋权的斯坦纳树

- 问题描述：给定若干个点和一个有权网格，通过斯坦纳树的思想将这些网格连线起来，使得连线经过的网格长度与加权值最小。但是在VLSI中连接的线是直角线，所以这些网格只能通过直角边连接，在建立斯坦纳树时，斯坦纳点往往出现在Hanan网格上，可以通过Hanan网格大幅度缩短解空间。
- 目标：
  - 网络加权值最小

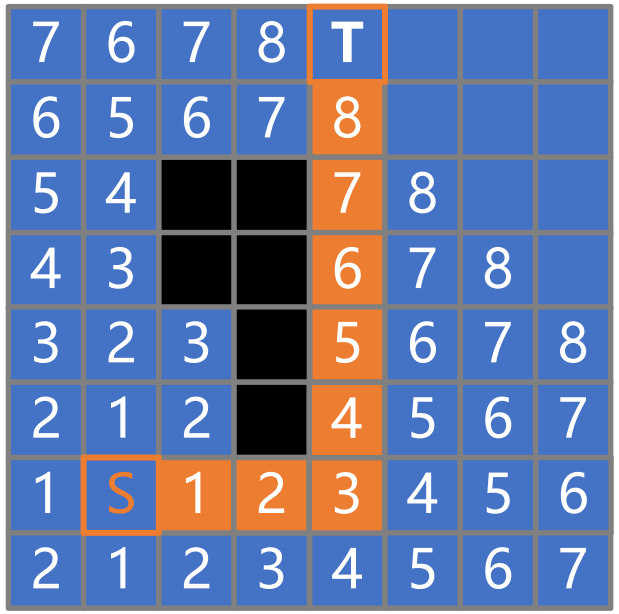
$$\text{Max} \sum_{i}^{net} \sum_{j}^{net} \text{weight}_{i,j}$$



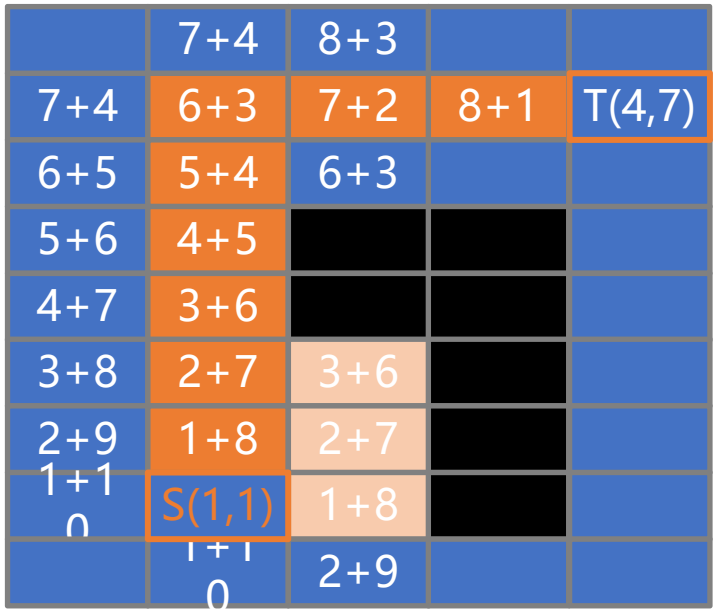
原路径（无权值影响斯坦纳树）　　权值影响下的新路径

两点布线

- 问题描述：将版图上的两点以最优的直角路径连接起来，同时还要躲避障碍。
- 目标：
  - 线长最短
- 输入：
  - 斯坦纳树库
  - 点坐标
- 输出：
  - 连接结果
- 约束：
  - 满足DRC



迷宫布线



A*布线

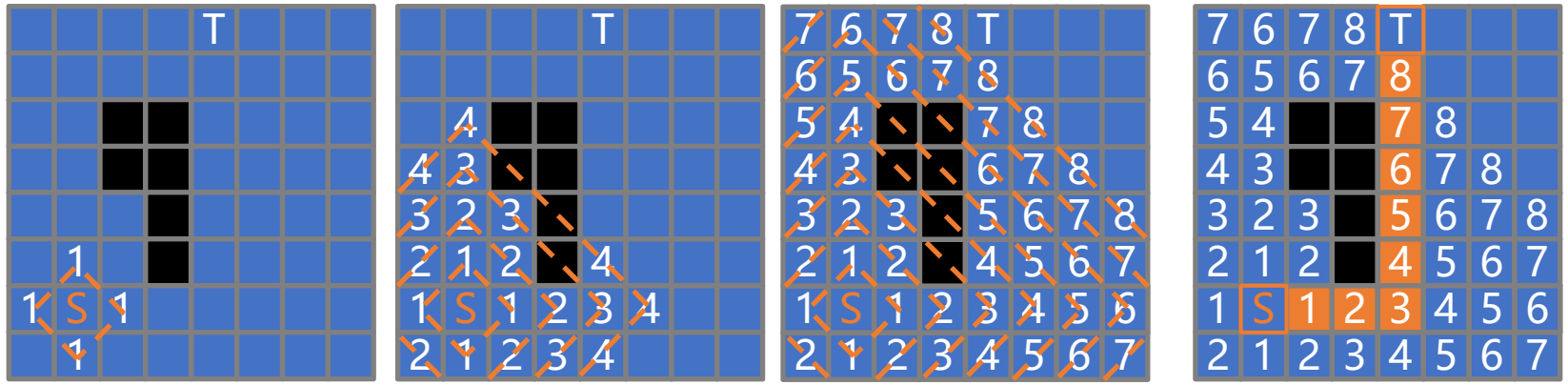
Maze迷宫布线

- 描述：lee迷宫布线算法是在net内选取一个点，这个为中心，以波的形式向外扩散，当波到达目标，即开始回溯。
- 优点：
  - 简单，灵活性高
- 缺点：
  - 变量多，运算时间大

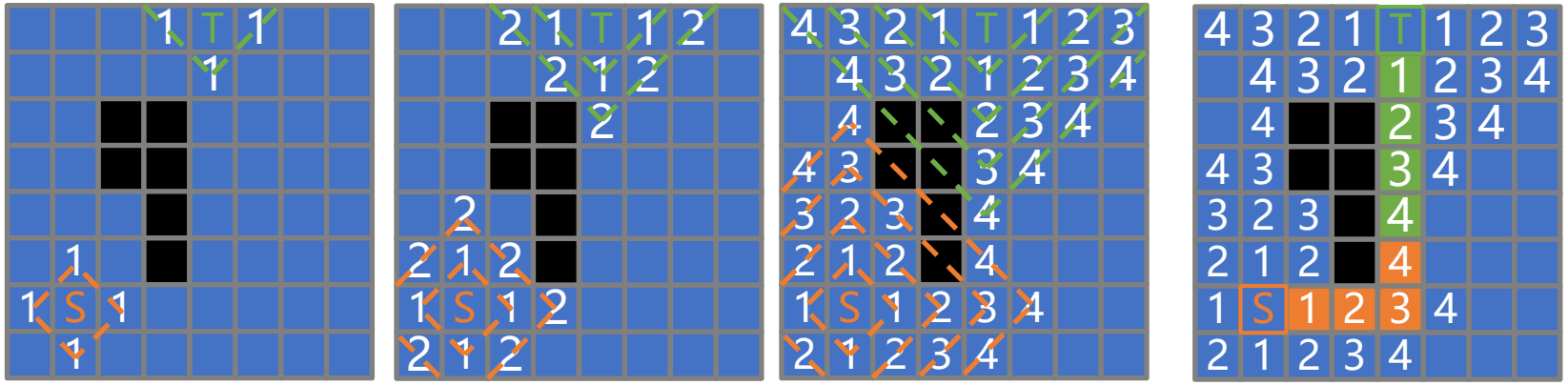
$$O(n) = |x_s - x_t| * |y_s - y_t|$$

  - 只能两点布线

单向阔波

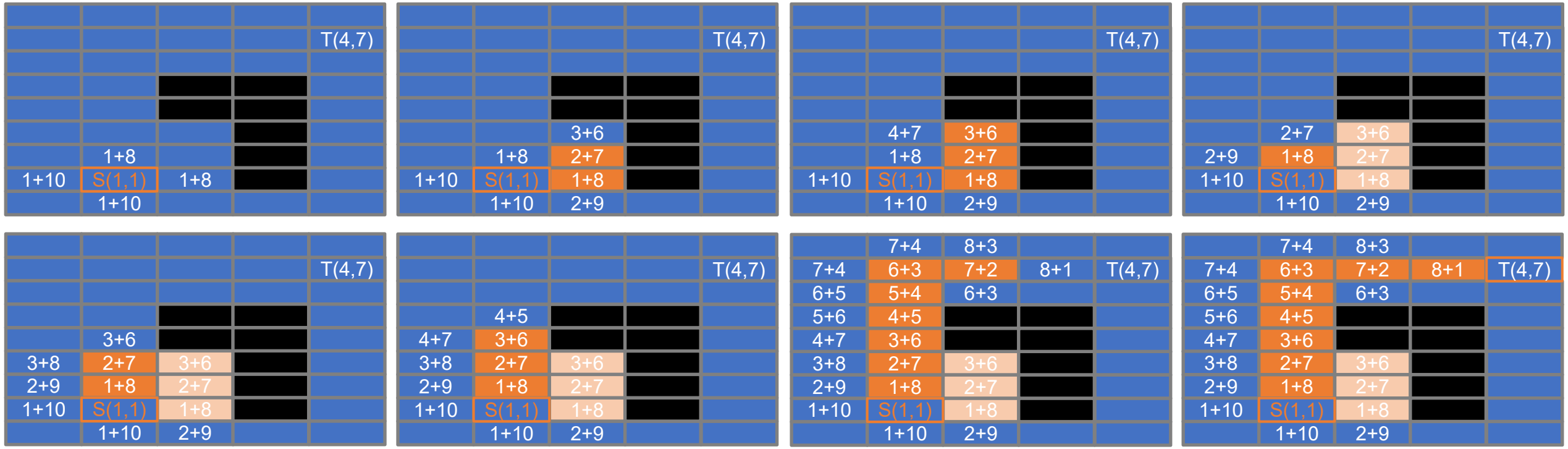


双向阔波



A*布线

- 描述：A*算法是在maze算法上增加了一定的指导性，在可以找到最佳路径的情况下，将时间复杂度降低。A*是一个启发式算法，其评估函数为f(n)=g(n)+h(n)，其中g(n)为起点到当前点的实际距离(不包括起点)，h(n)表示当前点到目标的估算距离(不包括当前点)，对于距离的计算，这里使用曼哈顿距离模型。



以上例子中的单位距离为1，每个框中的式子为g(n)+h(n)格式

加权图两点布线

- 问题描述：将版图上的两点以最短的直角路径连接起来，在保证权值最小的情况下，还要躲避障碍。
- 目标：
  - 线长最短
  - 加权值最小

$$\text{Min}\ \alpha * \text{wire\_length} + \beta * \sum_{i}^{net} \sum_{j}^{net} \text{weight}_{i,j}$$

- 输入：
  - 点坐标
  - 障碍信息
- 输出：
  - 连接结果
- 约束：
  - 满足DRC

0.2	0.1	0.3	0.5	0.7	0.7	0.8	T
0.1	0.2	0.8	0.5	0.5	0.9	0.7	0.3
0.5	0.4	0.8	0.7	0.5	0.1	0	0.2
0.1	0.1	0.2	0.1	0.1	0.6	0.8	0.1
0.3	0.5	0.1	0.2	0.1	0.5	0.3	0.1
0.7	0.9	0.4	0.3	0.7	0.1	0.4	0.3
0.6	0.1	0.6	0.7	0.4	0.4	0.1	0.5
S	0.7	0.4	0.2	0.3	0.4	0.7	0.8

加权迷宫布线

- 描述：lee迷宫布线算法是在net内选取一个点，这个为中心，以波的形式向外扩散，当波到达目标，即开始回溯。
- 优点：
  - 简单，灵活性高
- 缺点：
  - 变量多，运算时间大

$$O(n) = |x_s - x_t| * |y_s - y_t|$$

  - 只能两点布线

3	0.3	0.5	0.1	0.2	T
2	0.7	0.9	0.4	0.3	0.7
1	0.6	0.1	0.6	0.7	0.4
0	S	0.7	0.4	0.2	0.3
	0	1	2	3	4

				T
0.6下				
S	0.7左			

				T
1.3下				
0.6下	0.7左			
S	0.7左	1.1左		

1.6下	2.1左,下			T
1.3下	1.6下	1.7下		
0.6下	0.7左	1.3左	2.0左,下	
S	0.7左	1.1左	1.3左	1.6左

1.6下	2.1左,下	1.8下		T
1.3下	1.6下	1.7下	2.0左	
0.6下	0.7左	1.3左	2.0左,下	2.0下
S	0.7左	1.1左	1.3左	1.6左

1.6下	2.1左,下	1.8下	2.0左	T
1.3下	1.6下	1.7下	2.0左	2.7左,下
0.6下	0.7左	1.3左	2.0左,下	2.0下
S	0.7左	1.1左	1.3左	1.6左

1.6下	T	1.8下	2.0左	T
1.3下	1.6下	1.7下	2.0左	2.7左,下
0.6下	0.7左	1.3左	2.0左,下	2.0下
S	0.7左	1.1左	1.3左	1.6左

权值A*布线

- 问题描述：将版图上的两点以最短的直角路径连接起来，在保证权值最小的情况下，还要躲避障碍。
- 目标：
  - 线长最短
  - 加权值最小

0.3	0.5	0.1	0.2	T
0.7	0.9	0.4	0.3	0.7
0.6	0.1	0.6	0.7	0.4
S	0.7	0.4	0.2	0.3

				T
0.6+0.5				
S	0.7+0.5			

				T
1.3+0.4				
0.6+0.5	0.7+0.4			
S	0.7+0.5			

				T
1.3+0.4	1.6+0.3			
0.6+0.5	0.7+0.4	1.3+0.3		
S	0.7+0.5			

				T
1.3+0.4	1.6+0.3			
0.6+0.5	0.7+0.4	1.3+0.3		
S	0.7+0.5	1.1+0.4		

				T
1.3+0.4	1.6+0.3			
0.6+0.5	0.7+0.4	1.3+0.3		
S	0.7+0.5	1.1+0.4	1.3+0.3	

				T
1.3+0.4	1.6+0.3			
0.6+0.5	0.7+0.4	1.3+0.3	2+0.2	
S	0.7+0.5	1.1+0.4	1.3+0.3	1.6+0.2

				T
1.3+0.4	1.6+0.3	1.7+0.2		
0.6+0.5	0.7+0.4	1.3+0.3	2+0.2	
S	0.7+0.5	1.1+0.4	1.3+0.3	1.6+0.2

1.6+0.3				T
1.3+0.4	1.6+0.3	1.7+0.2		
0.6+0.5	0.7+0.4	1.3+0.3	2+0.2	2+0.1
S	0.7+0.5	1.1+0.4	1.3+0.3	1.6+0.2

1.6+0.3		1.8+0.1		T
1.3+0.4	1.6+0.3	1.7+0.2	2+0.1	
0.6+0.5	0.7+0.4	1.3+0.3	2+0.2	2+0.1
S	0.7+0.5	1.1+0.4	1.3+0.3	1.6+0.2

1.6+0.3	2.3+0.2	1.8+0.1	2+0	T
1.3+0.4	1.6+0.3	1.7+0.2	2+0.1	
0.6+0.5	0.7+0.4	1.3+0.3	2+0.2	2+0.1
S	0.7+0.5	1.1+0.4	1.3+0.3	1.6+0.2

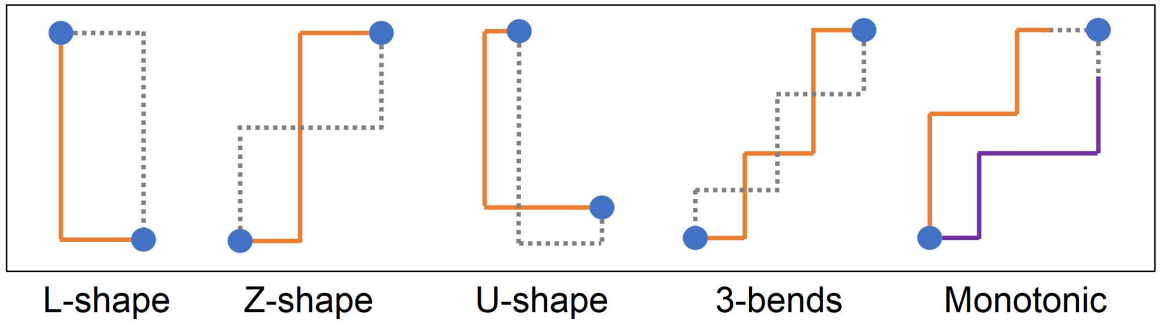
1.6+0.3	2.1+0.2	1.8+0.1	2+0	T
1.3+0.4	1.6+0.3	1.7+0.2	2+0.1	
0.6+0.5	0.7+0.4	1.3+0.3	2+0.2	2+0.1
S	0.7+0.5	1.1+0.4	1.3+0.3	1.6+0.2

1.6+0.3	2.1+0.2	1.8+0.1	2+0	T
1.3+0.4	1.6+0.3	1.7+0.2	2+0.1	
0.6+0.5	0.7+0.4	1.3+0.3	2+0.2	2+0.1
S	0.7+0.5	1.1+0.4	1.3+0.3	1.6+0.2

1.6+0.3	2.1+0.2	1.8+0.1	2+0	T
1.3+0.4	1.6+0.3	1.7+0.2	2+0.1	
0.6+0.5	0.7+0.4	1.3+0.3	2+0.2	2+0.1
S	0.7+0.5	1.1+0.4	1.3+0.3	1.6+0.2

模式布线

- - L型：两条直角连接线，适合连接相对简单的两点布线。
- - Z型：多用于需要拐弯的场景，但保持了较少的拐点数量。
- - U型：用于需要绕过障碍的简单情况，但仍然控制了路径长度。
- - 3-拐弯布线：允许多达三次转折，能够绕过少量障碍，提高了灵活性。
- - 动态模式布线：通过扩展基本模式布线的搜索空间，

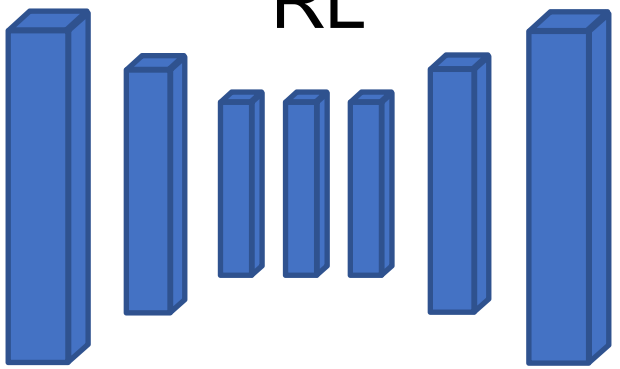


强化学习下的两点布线

iEDA

input feature

0.2	0	0.3	0.5	0.7	0.7	0.8	T
0	0.2	0.8	0.5	0.5	0.9	0.7	0.3
0.5	0.4	0.8	0.7	0.5	0.1	0	0.2
0.1	0.1	0.2	0.1	0.1	0.6	0.8	0.1
0.3	0.5	0	0.2	0.1	0.5	0.3	0
0.7	0.9	0.4	0.3	0.7	0.1	0.4	0.3
0.6	0.1	0.6	0.7	0.4	0.4	0.1	0.5
S	0.7	0.4	0.2	0.3	0.4	0.7	0.8

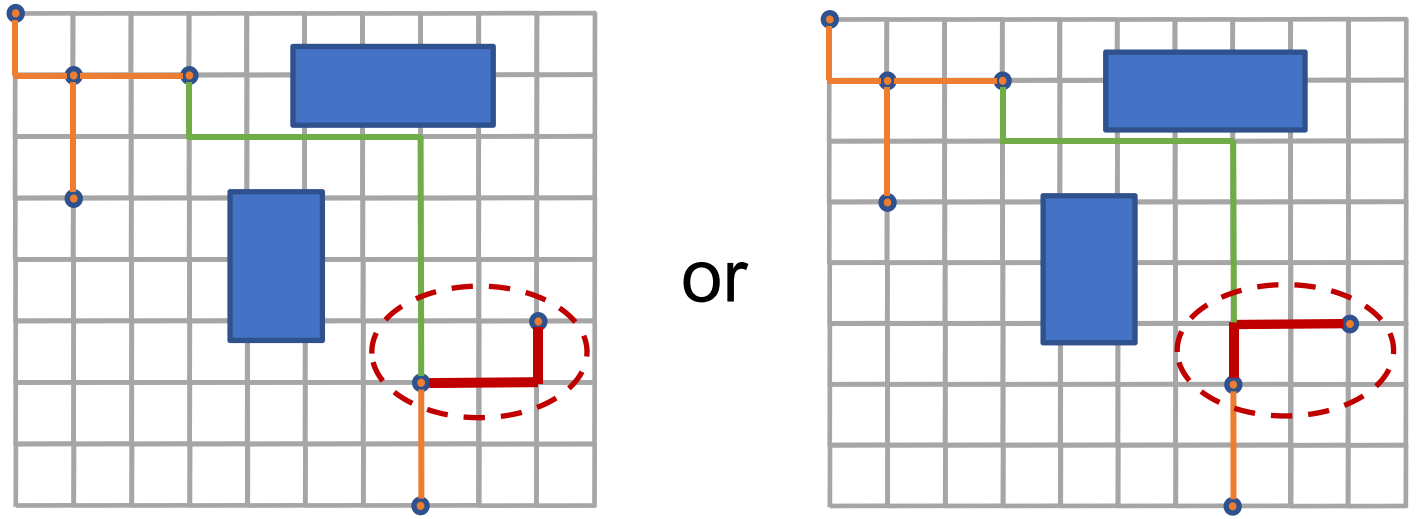


output

0.2	0	0.3	0.5	0.7	0.7	0.8	T
0	0.2	0.8	0.5	0.5	0.9	0.7	0.3
0.5	0.4	0.8	0.7	0.5	0.1	0	0.2
0.1	0.1	0.2	0.1	0.1	0.6	0.8	0.1
0.3	0.5	0	0.2	0.1	0.5	0.3	0
0.7	0.9	0.4	0.3	0.7	0.1	0.4	0.3
0.6	0.1	0.6	0.7	0.4	0.4	0.1	0.5
S	0.7	0.4	0.2	0.3	0.4	0.7	0.8

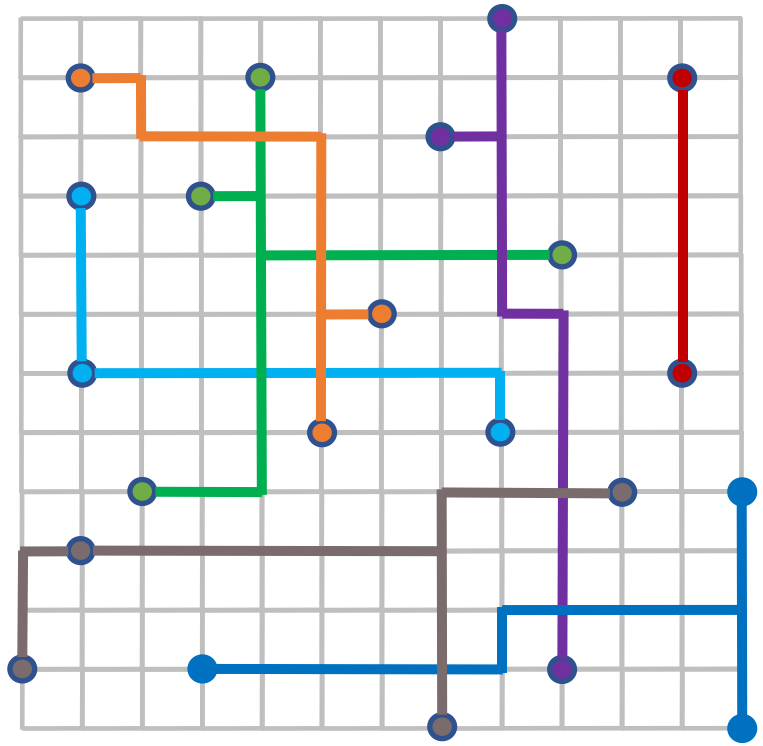
多线网布线

- 问题描述：每一个net内的pin的分布并不总是理想的，会因为布局的结果，有时会出现两个或多个点群的情况，需要将这些网群连接起来，同时还要躲避障碍。



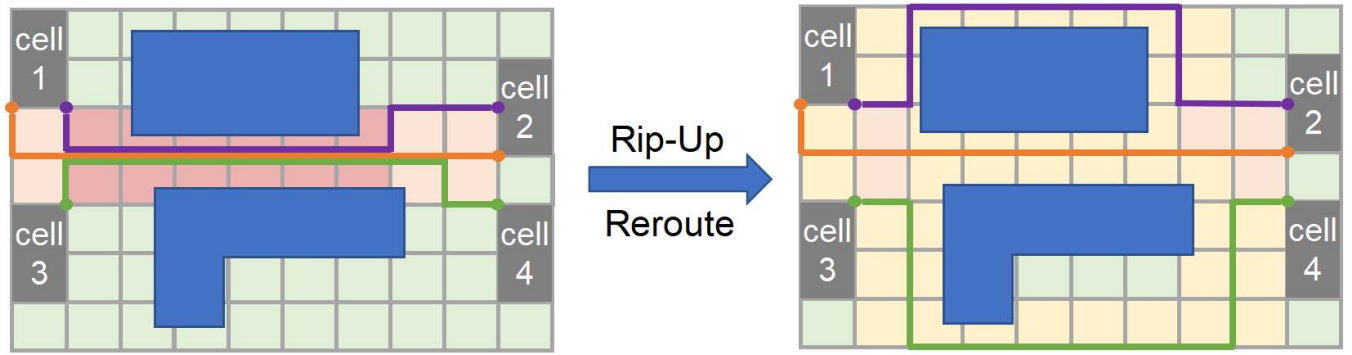
多线网布线

- 顺序布线
  - 将网络按预定的顺序布线。例如，先布线网络A，再布线网络B，然后布线网络C。
- 并发布线
  - 多商品流：
    - 1）将布线问题建模为多商品流问题。网络图 G = (V, E) 中的每条边 e ∈ E 具有流量容量 cap(e) 和成本 cost(e)。
  - 整数规划布线
    - 用于表示每个线网的可选布线方案。对于每个线网 （是线网集合），有 个布线方案，每个选项由布尔变量 控制。如果 ，表示该线网选用了这个布线方案；如果 ，表示该方案未被选用。
- 协商布线
  - 协商布线的核心思想是利用布线图中每条边的拥塞历史信息，作为未来布线的基础。主要目标是：
  - - 平衡拥塞消除：确保布线过程中不会出现过度拥塞，从而避免布线失败或电路性能下降。
  - - 最小化性能下降：在布线过程中，尤其是在时序关键路径上，尽量减少对性能的影响。



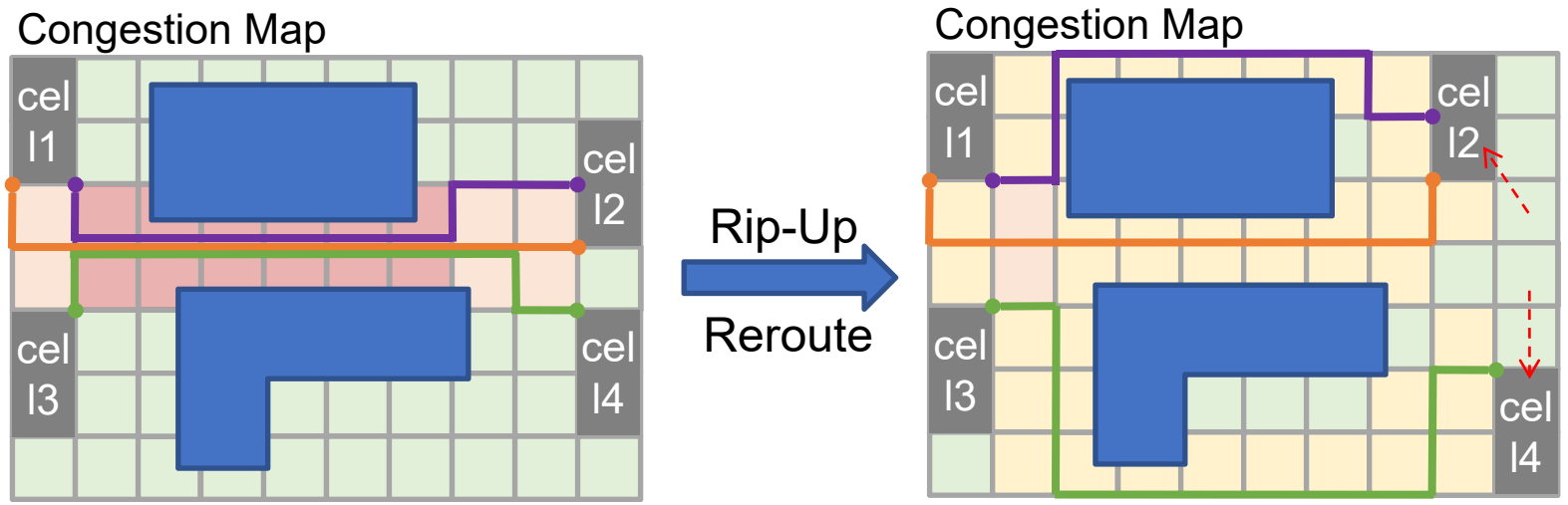
拆线重布

- 拆线重布线是一种用于解决布线资源过度使用问题的技术。拆线重布线的过程如下：
  - 1）初始布线：首先对一组线网进行初始布线。在这一阶段，系统可能会临时允许布线资源的违例，即可能会让多个线网共享同一布线资源或路径。这一步的目标是快速得到一个可行的初始布线方案。
  - 2）违例检测：在初始布线完成后，系统会检测是否存在资源违例的情况，例如某些布线路径过度拥塞或被多个线网占用，超出物理资源的容量。
  - 3）拆除违例线网：一旦检测到资源违例，系统将会选择违例的线网，将其从原有路径上拆除。这一步的目的是为后续的重新布线释放资源。
  - 4）重新布线：在拆除违例线网后，系统将对这些线网重新进行布线，尝试为它们找到新的路径。重新布线会尽量避开原先拥塞的区域，寻找资源未被过度使用的路径。
  - 5）迭代优化：上述过程会不断重复，逐步优化布线方案，直到所有的线网都能成功布线，并且没有资源违例或冲突为止。



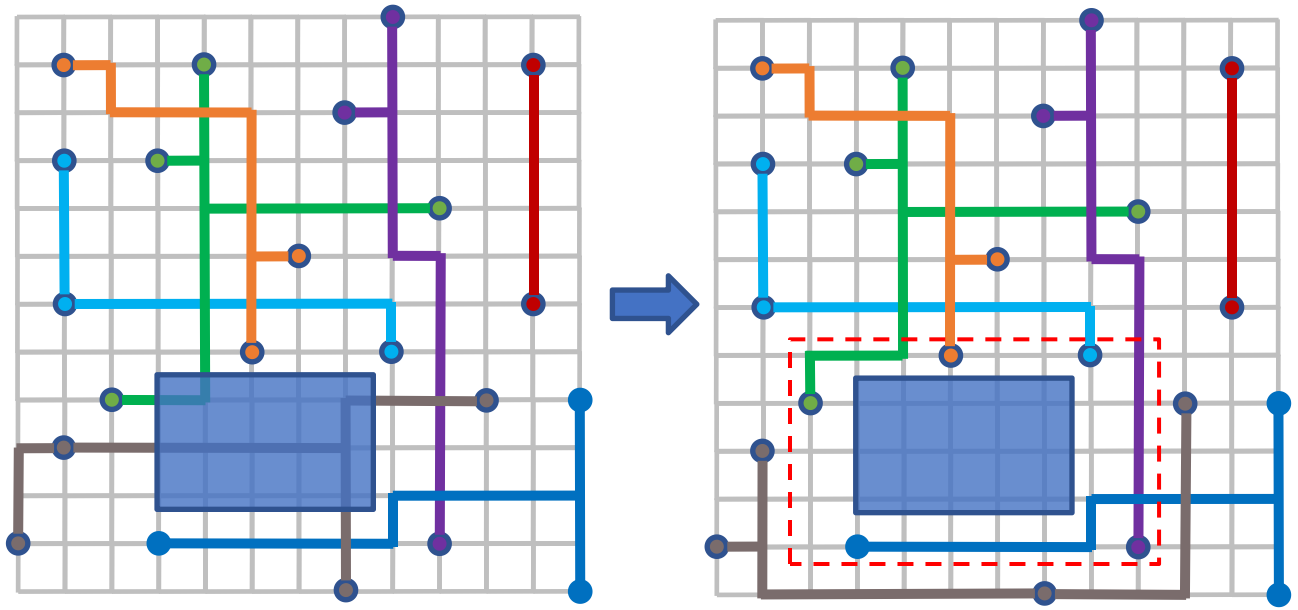
单元可移动的拆线重布

- 问题描述：虽然对于一个设计来说，全局布线的资源是一定的(假设设计给的总资源是足够的)，但是往往由于某些方法的局限性，导致局部资源不满足布线的要求，这就需要进行拆线重布，一般的拆线重布是改变线的连接位置，现在可以通过移动单元来影响pin的分布，但是一个单元不只是一个pin，移动单元会使得多个net发生变化，需要全局地考虑问题。
- 目标：
  - 使布线效果满足设计资源
  - 总的单元移动量最小



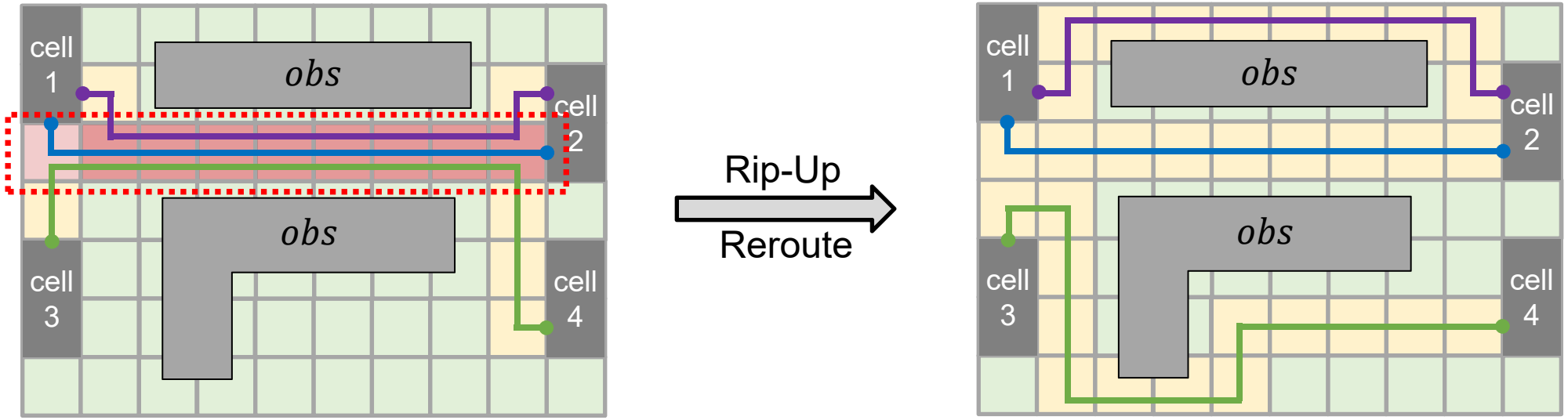
指定区域拆线重布

- 问题描述：当设计出现微小的改动(如在某些区域加入了一些器件)，可以不运行全局布线，而是指定对应的区域重做全局布线。



Routing Congestion

- When the number of resources required for routing exceeds the available resources within the GCell, it is considered that congestion has occurred in that GCell.



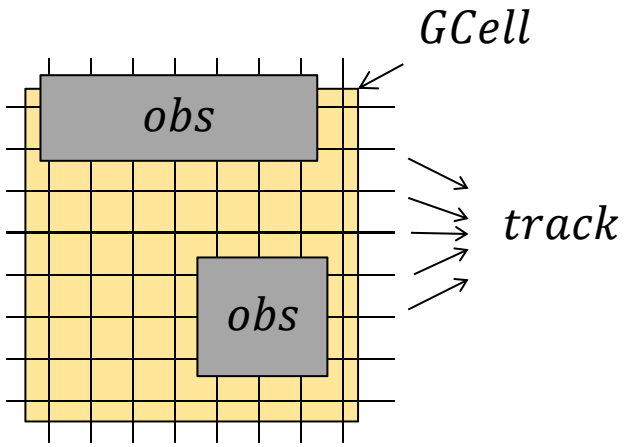
- Each GCell has one available resource, and when a routing passes through a GCell, it consumes one resource. Congestion is present at the red dashed box in the left diagram. After rip-up and reroute, as shown in the right diagram, the congestion is eliminated.

Routing Resource and Demand

- GCell Resource

$$r_i = Num(track_i) \cdot [1 - \frac{Area(obs_i)}{Area(g_i)}]$$

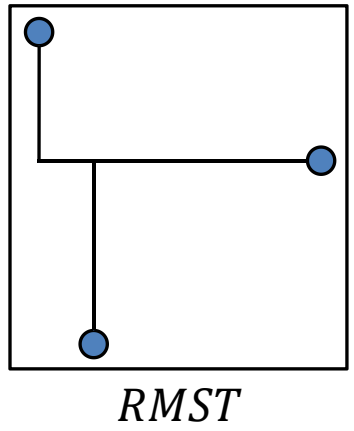
where r_i is the resource of the g_i (i-th GCell), which is calculated from the product of the number of tracks and the effective area ratio.



- Net Demand

$$d_j = WL(RSMT_j)$$

we use the length of rectilinear Steiner minimum tree (RSMT) as the demand for each net, which will serve as the lower bound for the final wire length of this net.



拥塞（congestion）和溢出（overflow）

$$congestion(e_{ij}) = \frac{u(e_{ij})}{c(e_{ij})}$$

- 其中，表示GCell 和之间的边的拥塞度，定义为与的比率 。容量 表示网格边的布线容量，是通过网格边的线网数。

其中，$congestion(e_{ij})$表示GCell G_i和G_j之间的边e_{ij}的拥塞度，定义为$u(e_{ij})$与$c(e_{ij})$的比率 。容量 $c(e_{ij})$ 表示网格边G_i的布线容量，$u(e_{ij})$是通过网格边G_i的线网数。

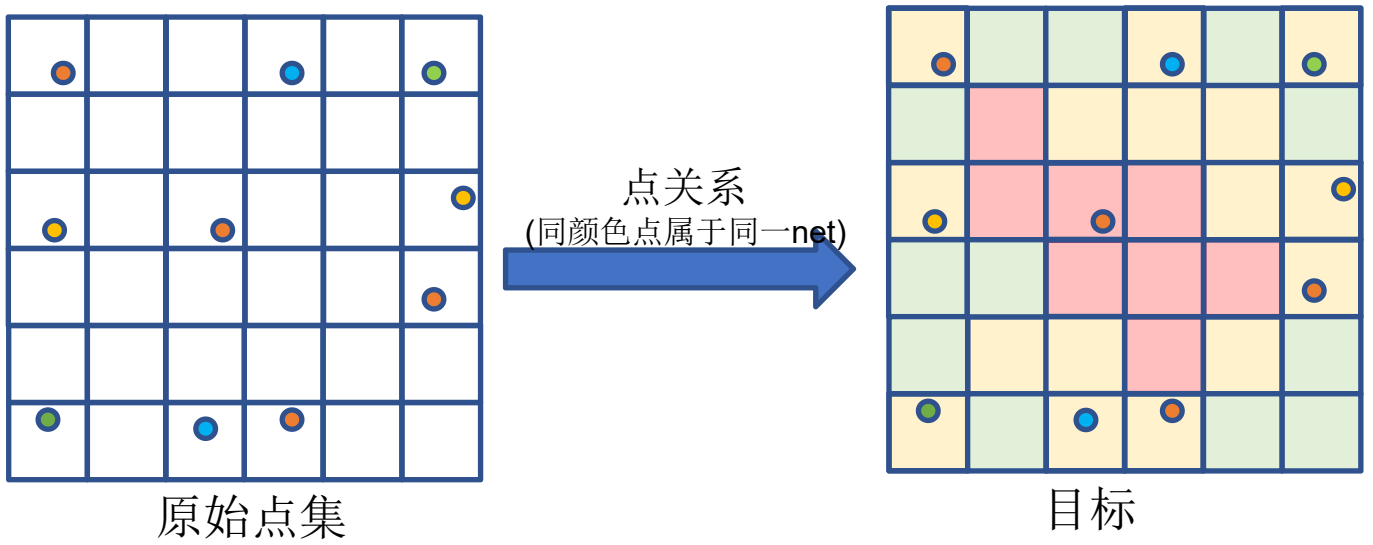
- 溢出数的数学表达式如下：

$$overflow(e_{ij}) = \begin{cases} u(e_{ij}) - c(e_{ij}), & \text{如果} u(e_{ij}) > c(e_{ij}) \\ 0, & \text{否则} \end{cases}$$

拥塞估计

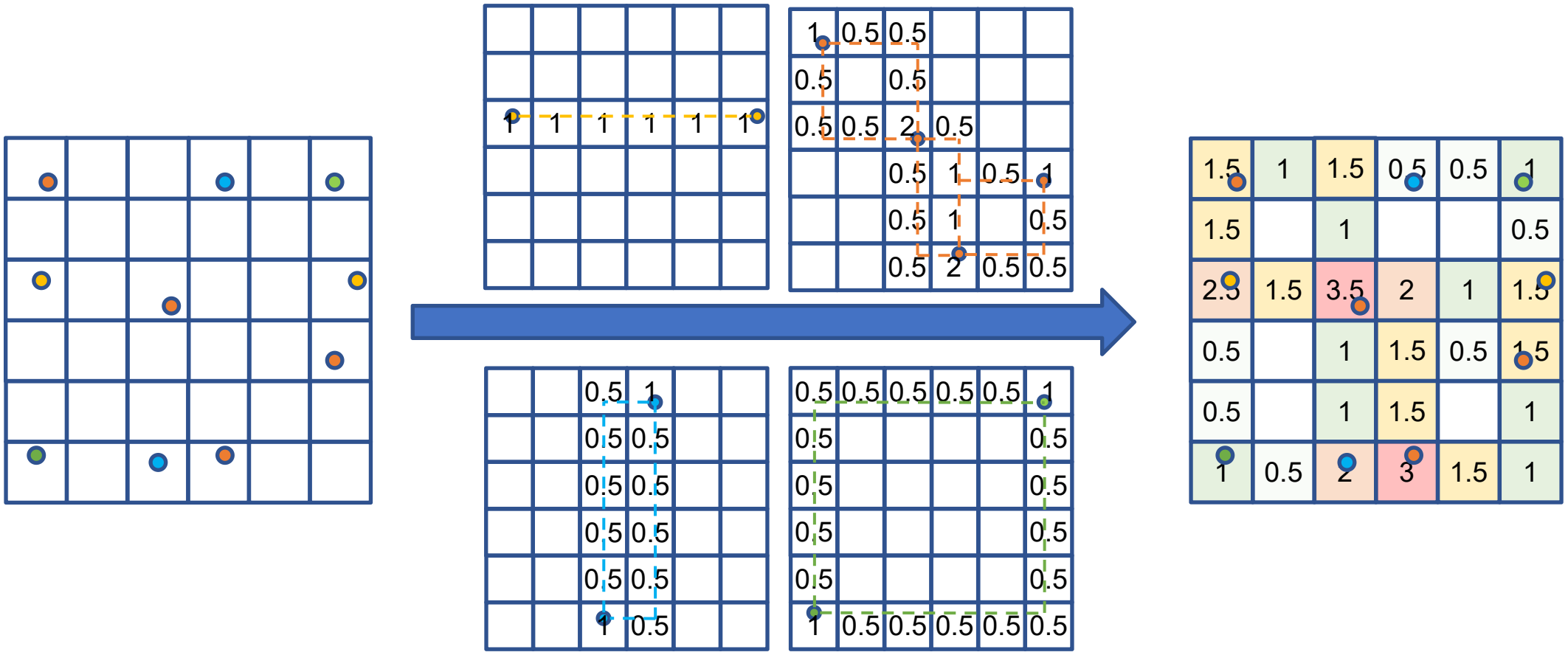
- 问题描述：在布线之前或过程中，需要对版图进行快速、准确的拥塞评估，版图内的pin都已经固定，通过这些pin的信息(哪些pin属于同一个net)，可以知道大致哪个区域会有绕线拥塞的出现，所以能预估得到对应的Congestion Map(CMap)，以更好地指导接下来的布线。
- 目标：
  - 最小化评估值与真实值之间的差

$$\mathrm{Min}\sum_{i}\sum_{j}(\mathrm{CMap}_{i,j}^{evaluate}-\mathrm{CMap}_{i,j}^{real})^2$$



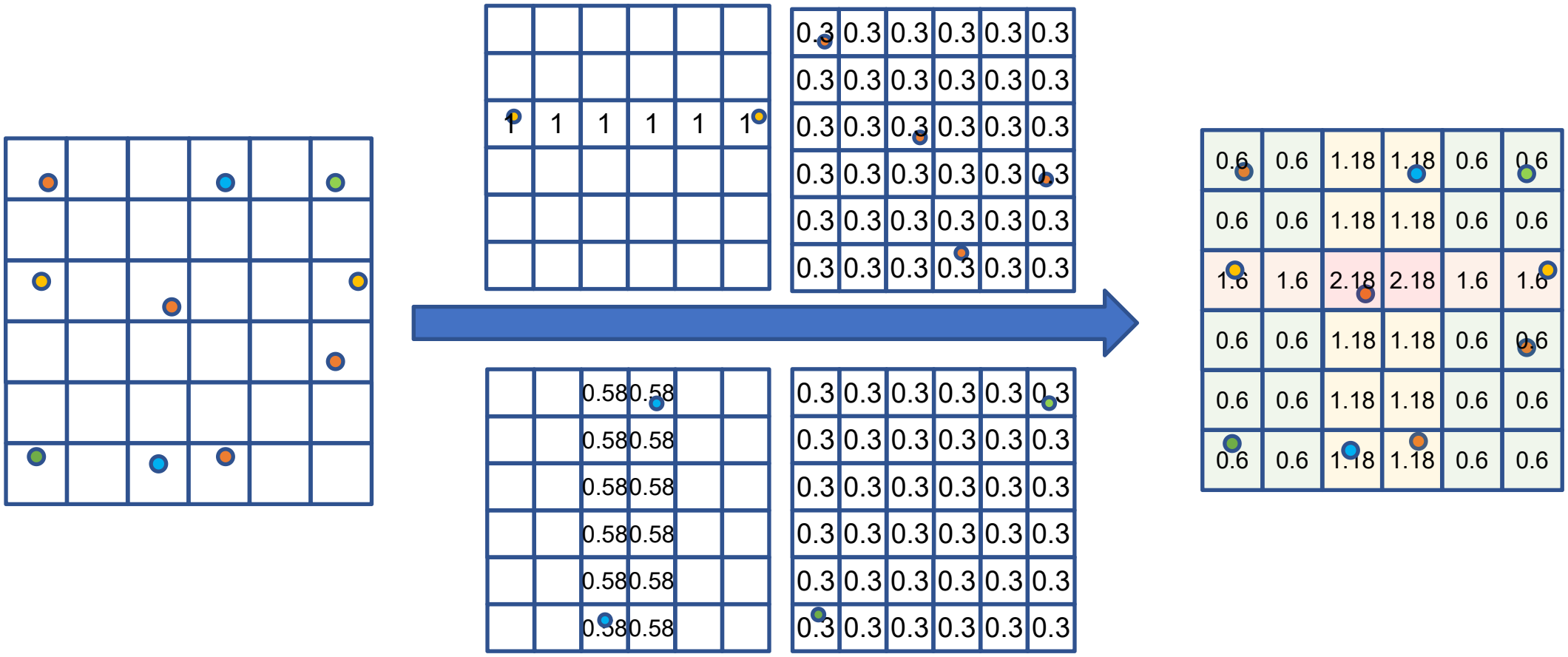
FastRoute

- 描述：将一个net划分为若干个two-pin线网，若每个two-pin线网pin的x坐标(y坐标)一样，则认为只能直连这样的一根线，将这根线经过的grid上添加上“1”的需求量，而对于x坐标(y坐标)不同的two-pin线网来说，相当于两个L型布线，每个L型布线经过的grid需求量加“0.5”。



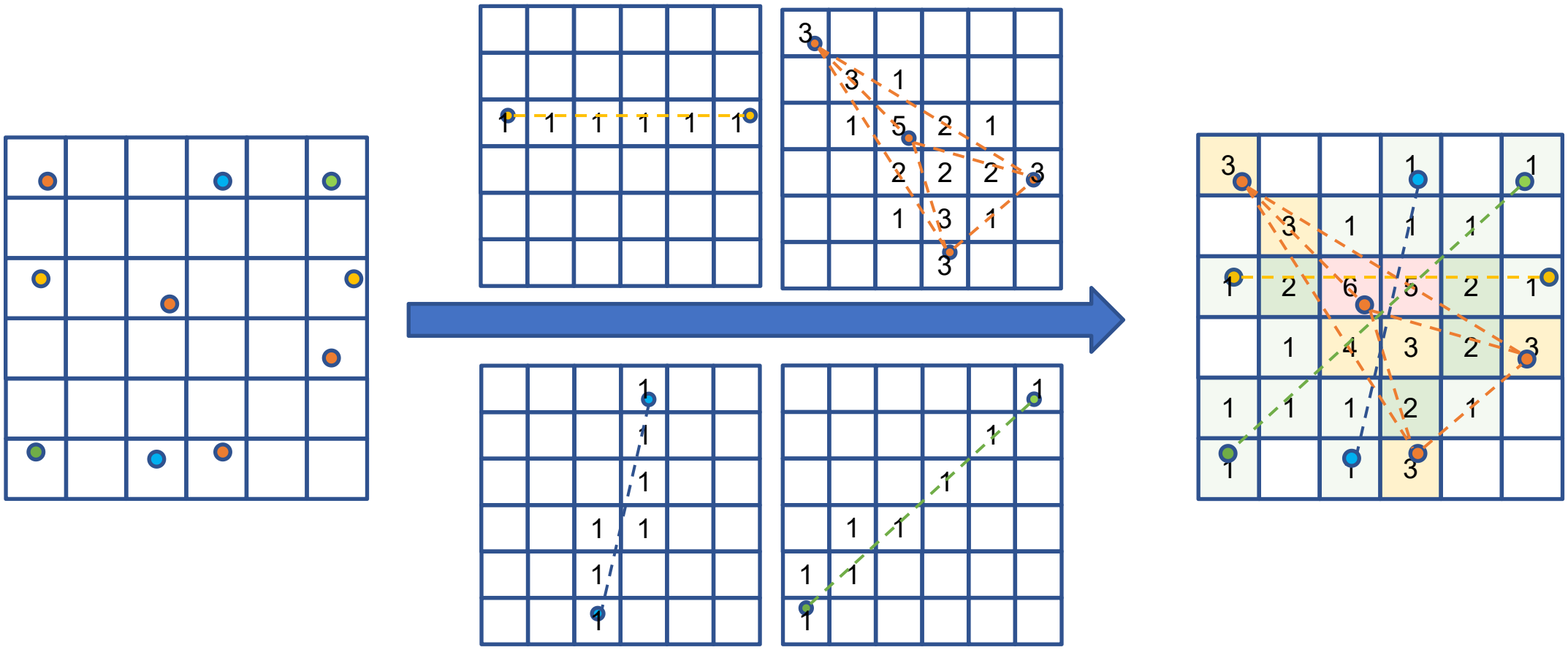
BBox初值重叠

- 描述：每个net都有对应的BBox，给每个BBox“着色”，然后求得全局叠加值，即为congestion Map。



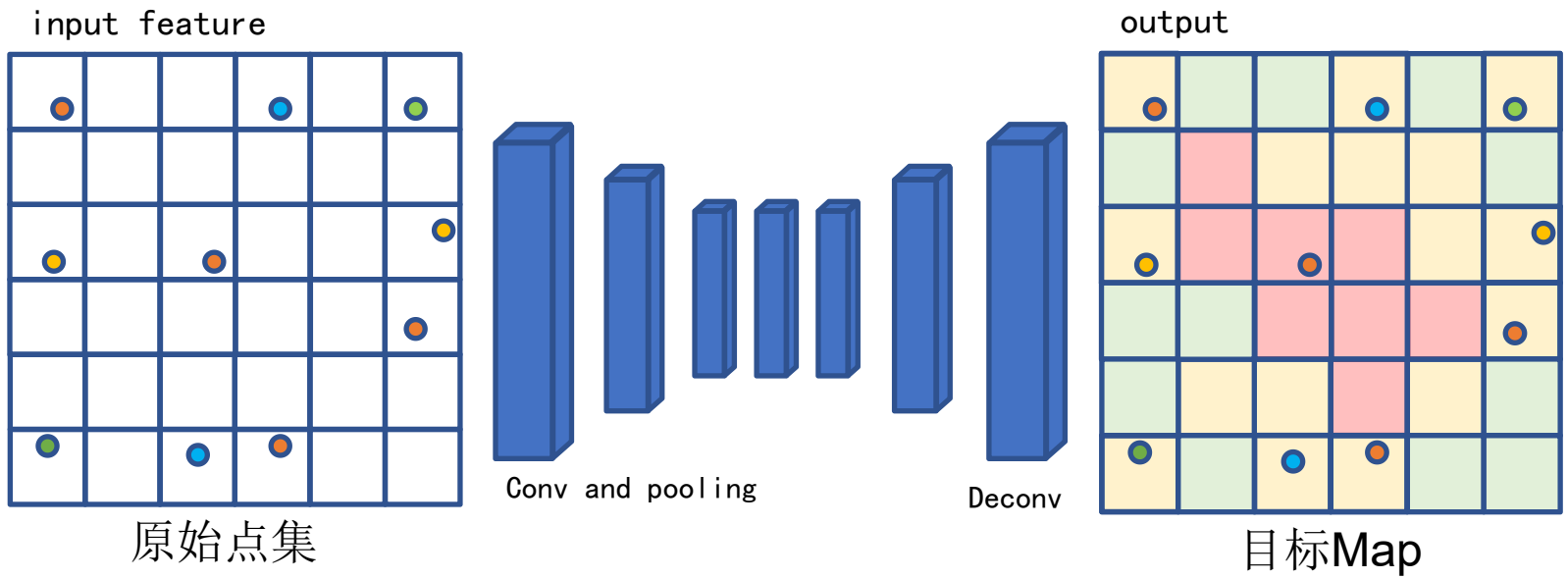
net虚连线估计

- 描述：将每个net的每个pin两两连接，虚连线经过的grid拥塞值加一。



机器学习下的Congestion估计

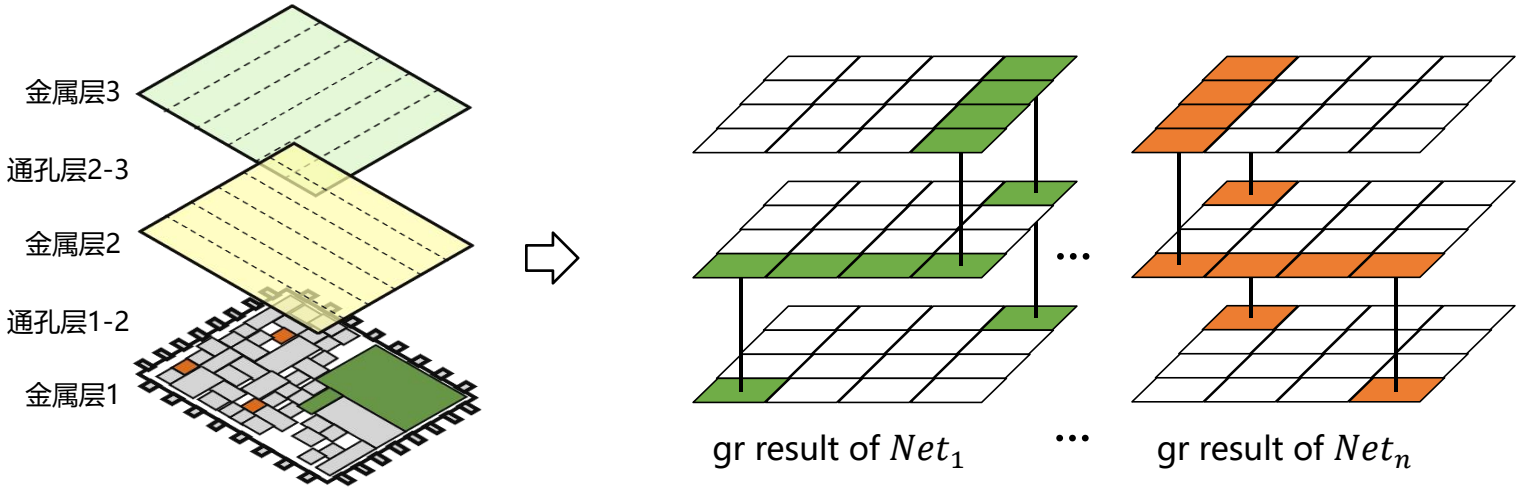
- 问题描述：在布线之前或过程中，需要对版图进行快速、准确的拥塞评估，版图内的pin都已经固定，通过这些pin的信息(哪些pin属于同一个net)，也许通过ML可以比传统算法更快地知道大致哪个区域会有绕线拥塞的出现，所以能预估得到对应的Congestion Map(CMap)，以更好地指导接下来的布线。

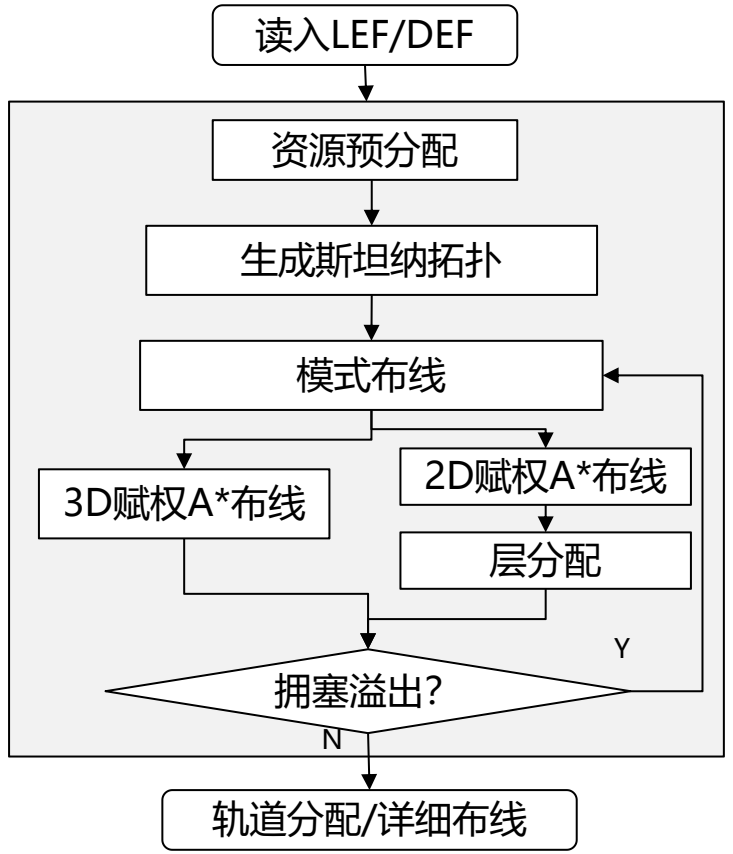


iEDA

全局布线

- 全局布线步骤
  - 资源分配
  - 斯坦纳树
  - 布线
  - 层分配

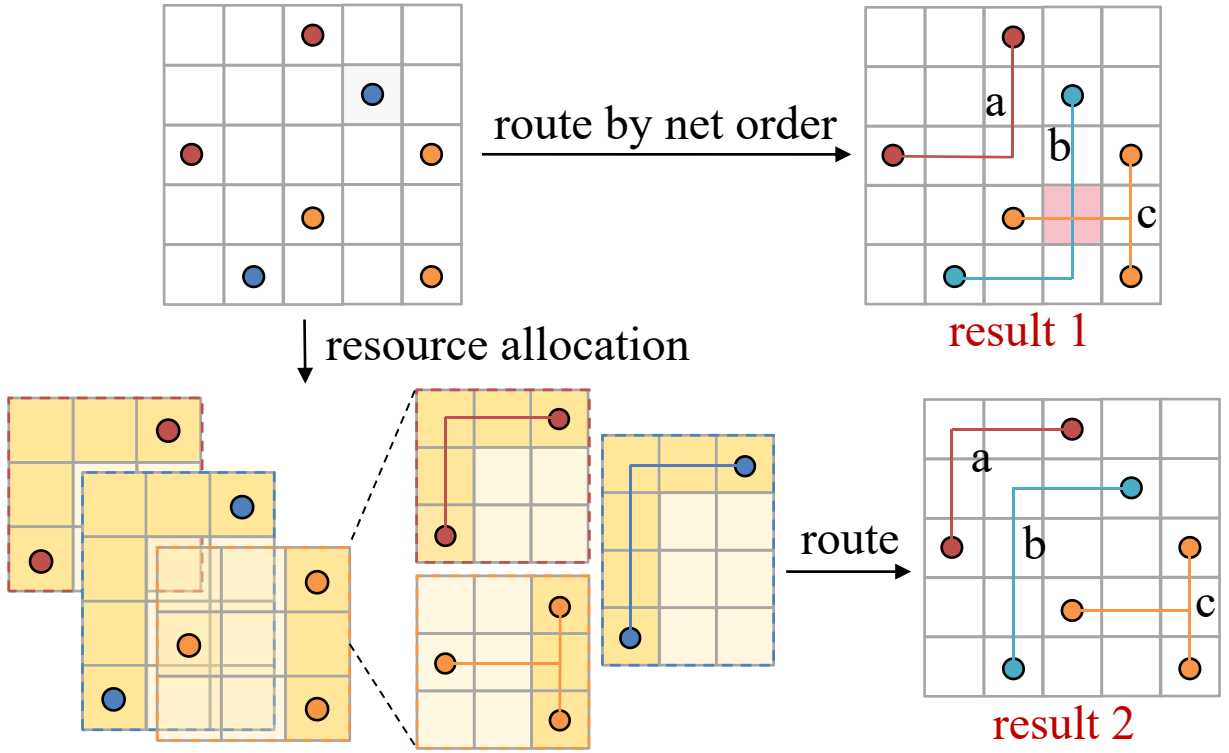




全局布线

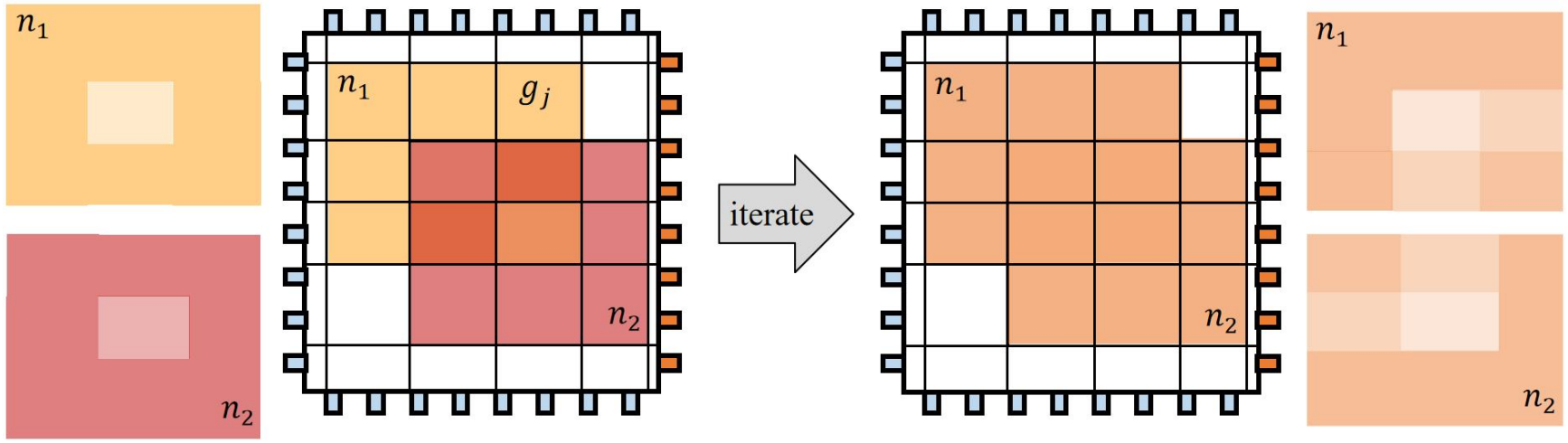
Resource Allocation

- Motivation
  - Most competitive routers heavily rely on net order, and repair operations after routing.
  - Provide a prior guidance for each net before routing.



Resource Allocation

- Iterative solution for Posterior Probability Density

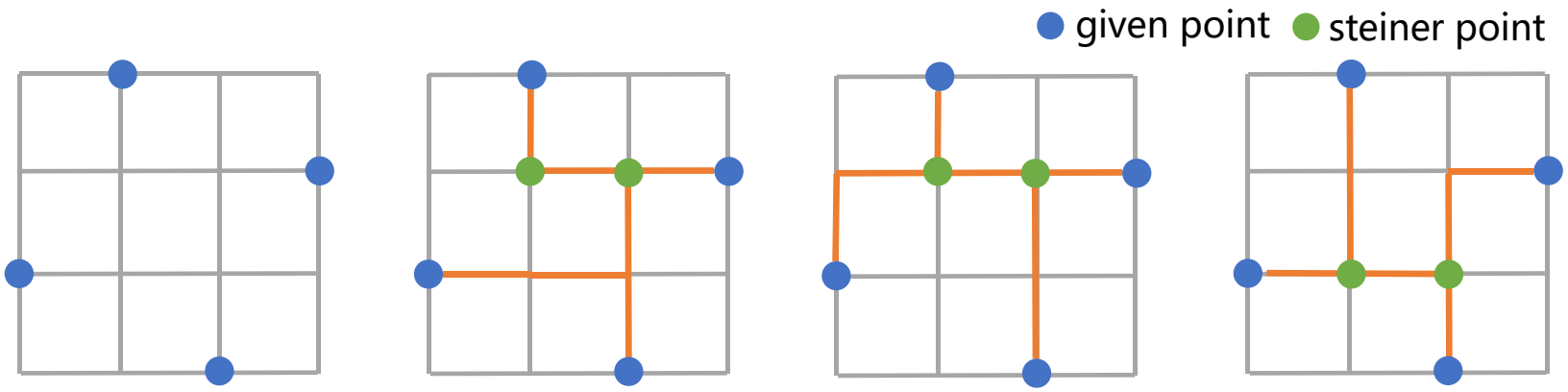


The global probability density is the superposition of the prior probability density of each net.

After iteration, the global probability density will become uniform, and each net will obtain its posterior probability density.

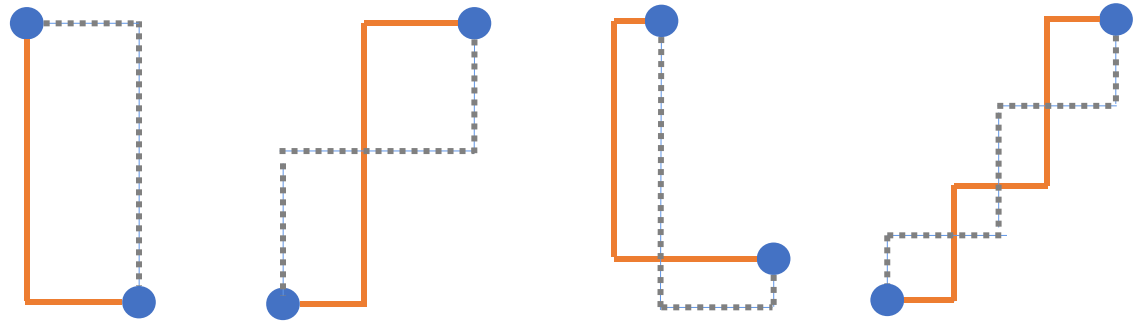
斯坦纳树

- Rectilinear Steiner tree (RST)
  - Given a set of points
  - Additional points (Steiner points) can be introduced
  - Limited to horizontal and vertical segments only

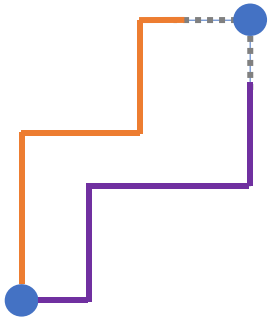


布线

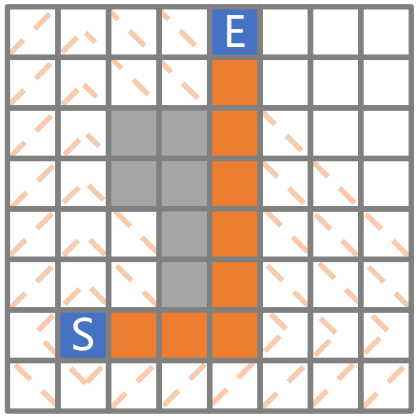
- Two Pin Routing
  - L, Z, U, 3-bends
  - Monotonic, Lee and A*
  - Multi-source Multi-sink



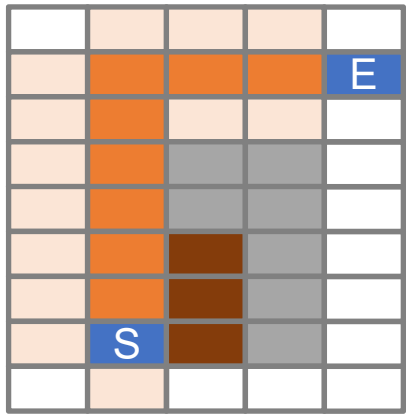
L-shape Z-shape U-shape 3-bends



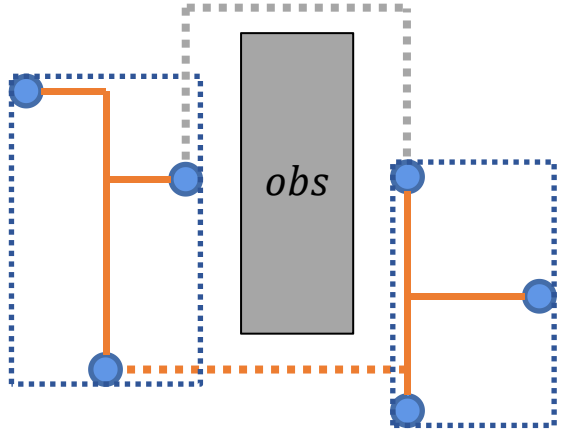
Monotonic



迷宫布线



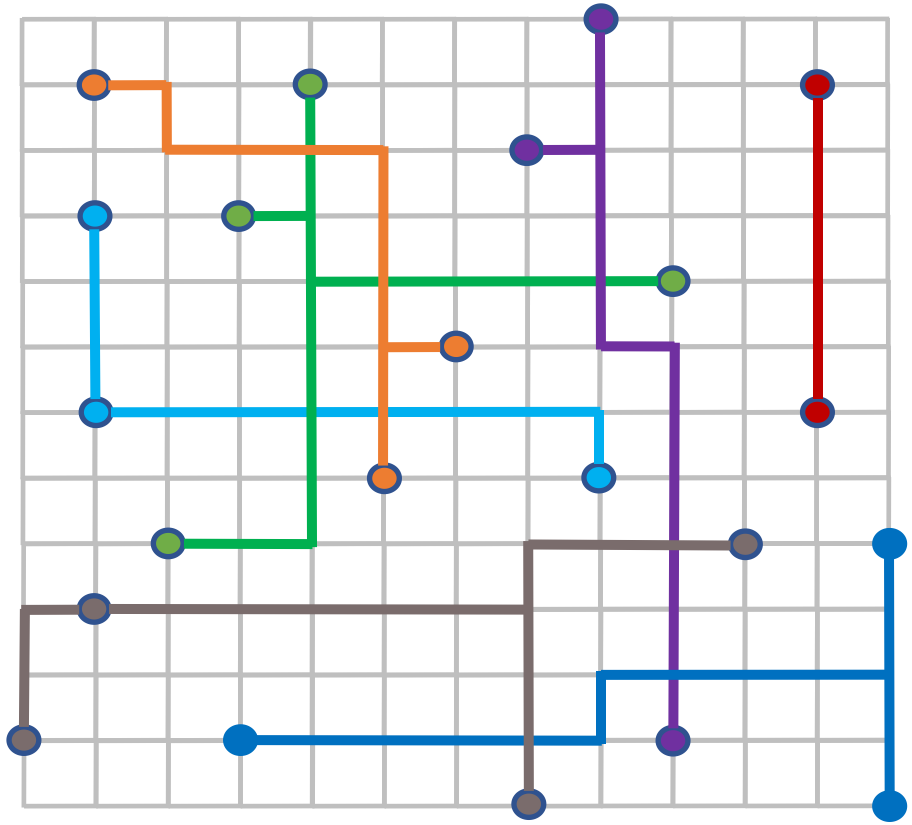
A*布线



Multi-source Multi-sink

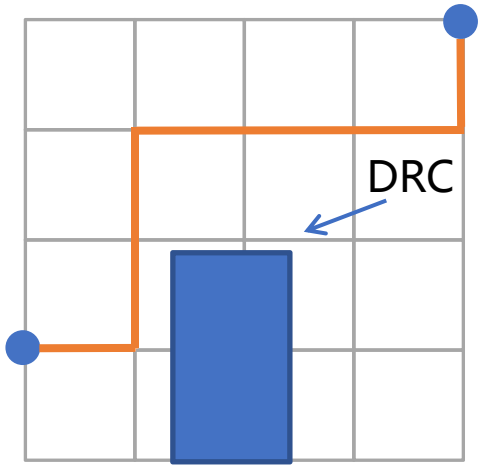
布线

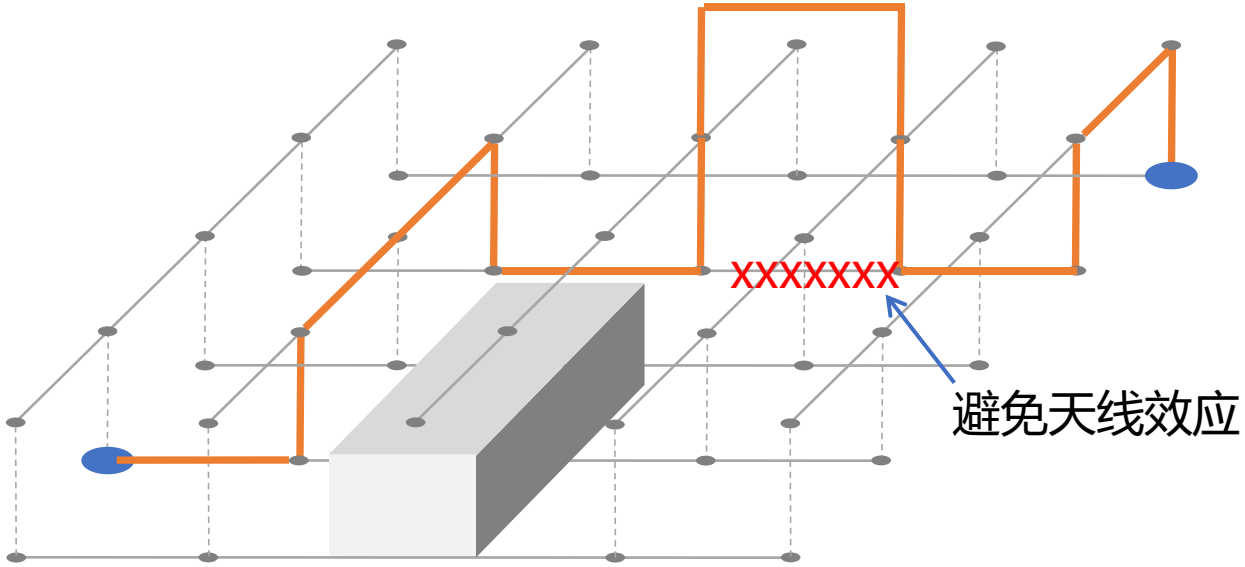
iEDA



层分配

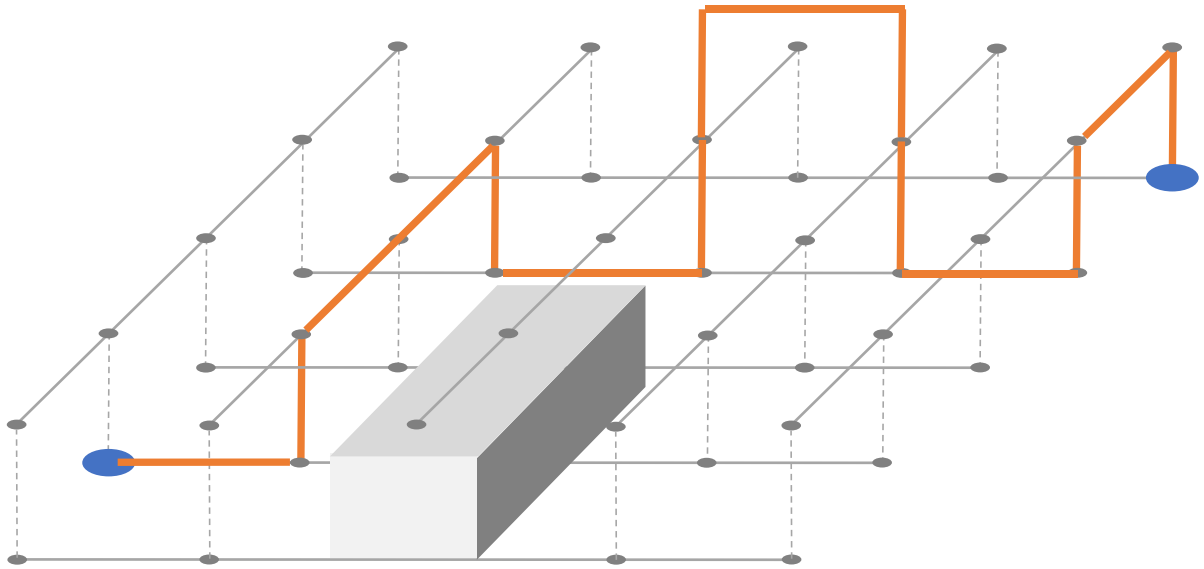
- 问题描述：全局布线通常分为两步，首先在二维上将线连接，再通过层分配将线升至高层。层分配不是简单地将每一段线分配一个层，还需要满足DRC，或者天线效应等制造需要，可能会将一段线进行划分。
- 目标：
  - 将线段分配到层上
  - 通孔数少





三维布线

- 问题描述：全局布线通常分为两步，首先在二维上将线连接，再通过层分配将线升至高层。本问题不同于之前，想要通过某种算法可以快速地在三维上直接布线，同样的，三维上也存在有不可布线的区域(macro或者人为规定的blockage)。
- 目标：
  - 连通
  - 线长最短
- 输入：
  - 三维点坐标
  - 障碍信息
- 输出：
  - 连接结果
- 约束：
  - 满足DRC



iEDA

01 Introduction

02 Routing Techniques

03 Global Routing

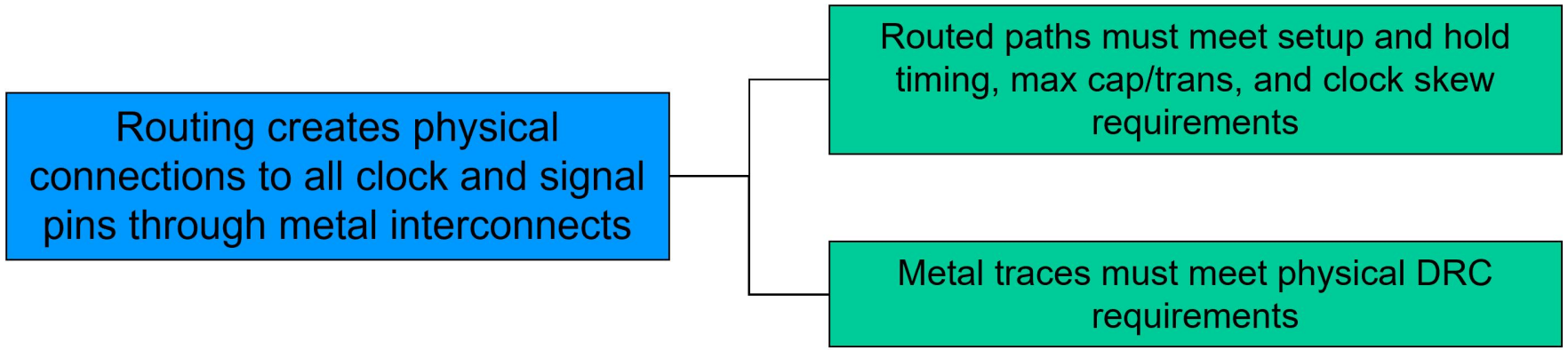
04 Detail Routing

05 Beyond Routing

Routing Fundamentals: Goal

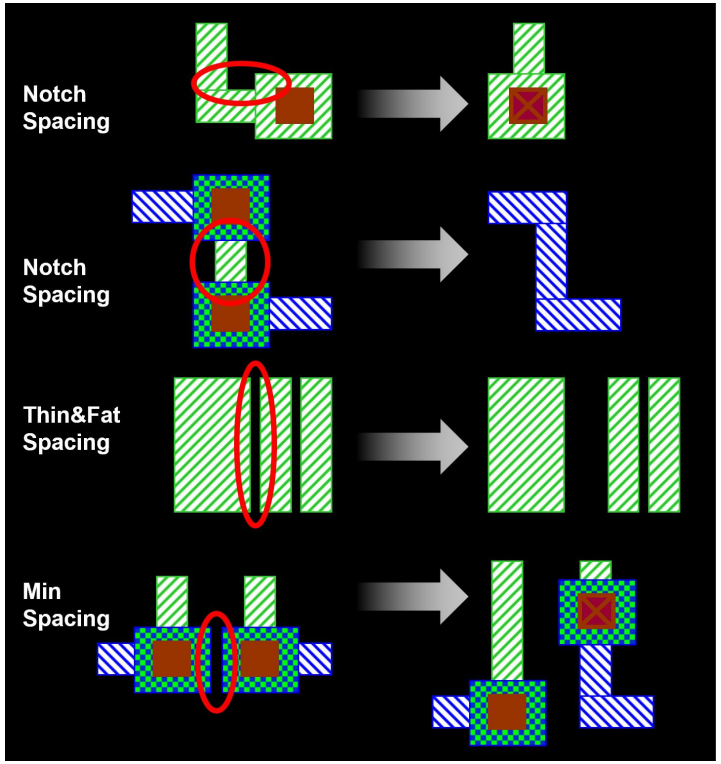
iEDA

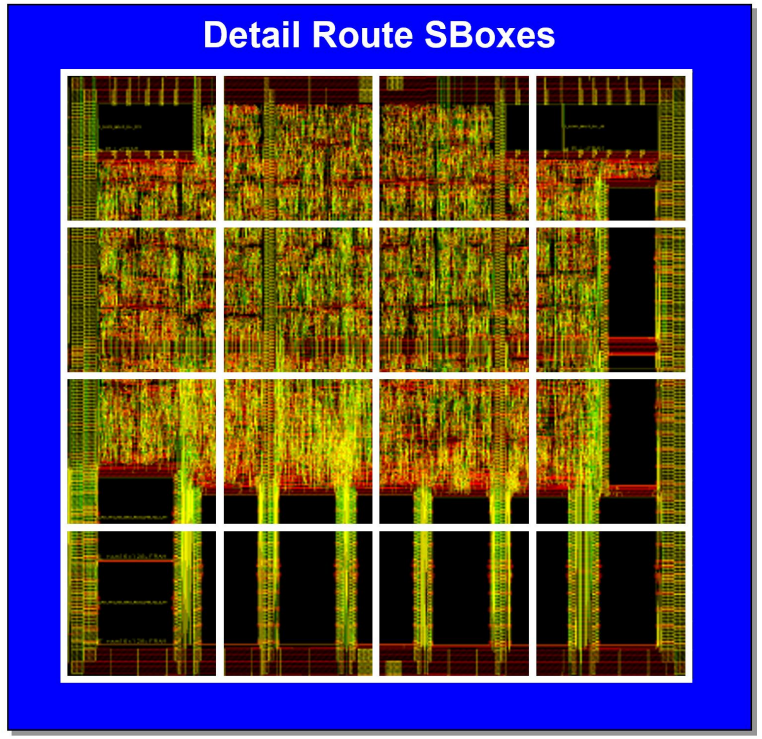
- Routing creates physical connections to all clock and signal pins through metal interconnects
  - Routed paths must meet setup and hold timing, max cap/trans, and clock skew requirements
  - Metal traces must meet physical DRC requirements



Route Operations: Detail Routing

- Detail route attempts to clear DRC violations using a fixed size Sbox
- Due to the fixed Sbox size, detail route may not be able to clear all DRC violations

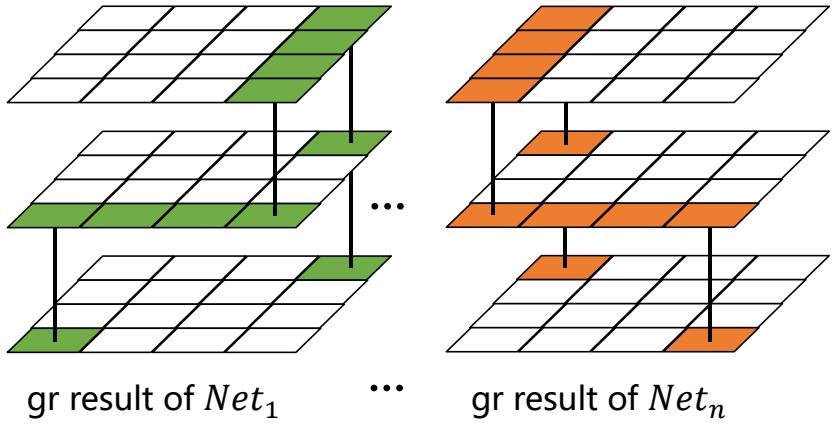


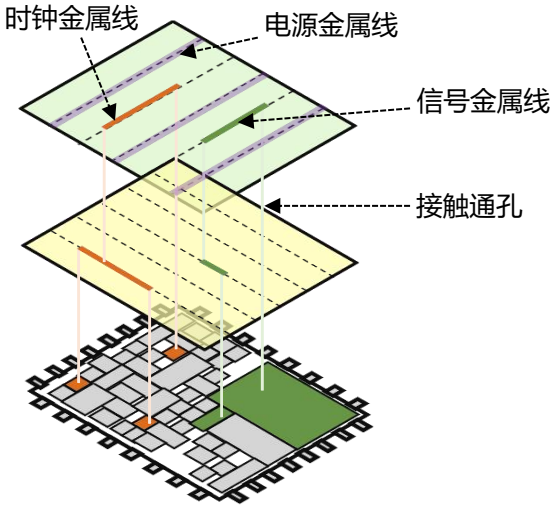


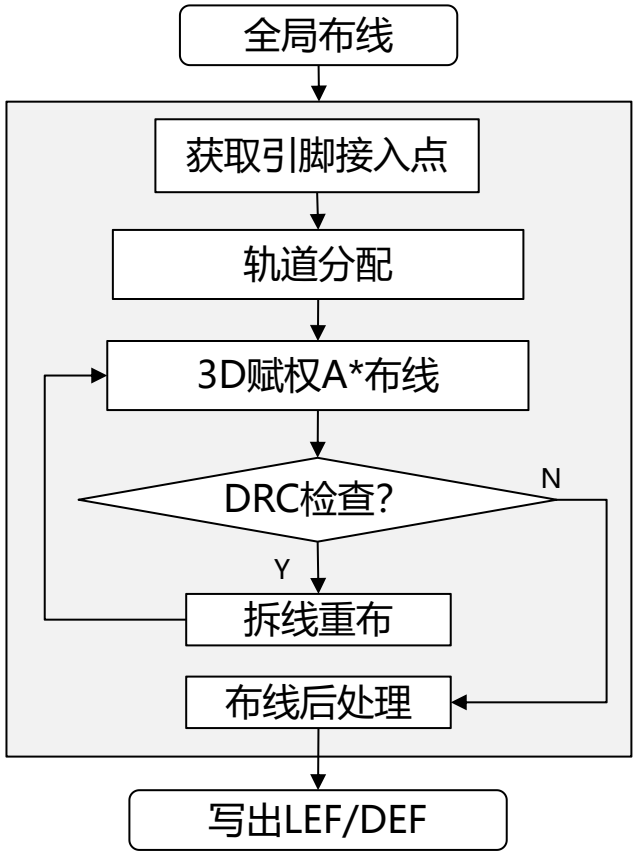
详细布线

详细布线步骤

- 引脚接入
- 轨道分配
- 布线
- 设计规则检查
- 拆线重布



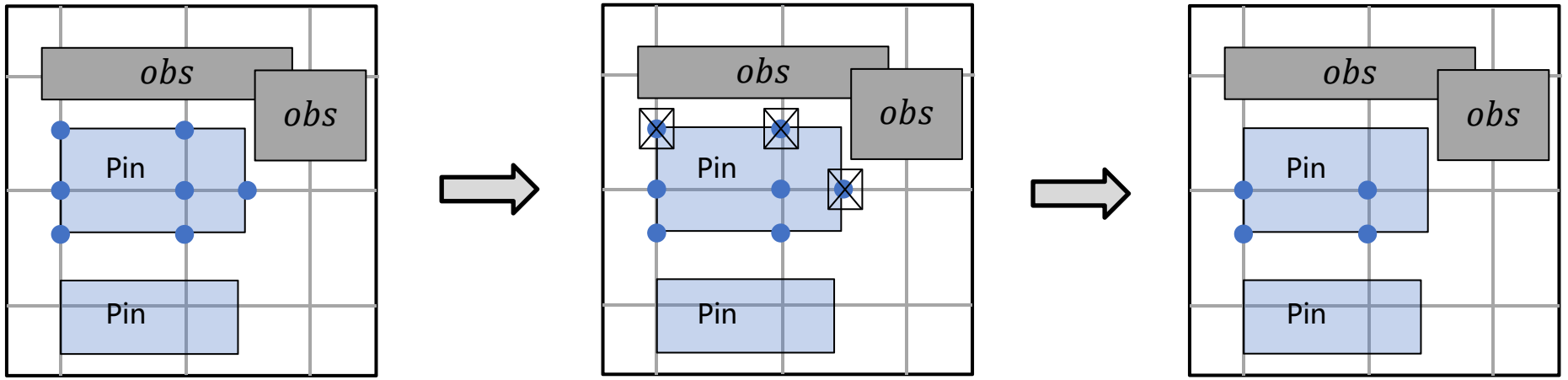




详细布线

Pin Access

- Generate valid access points
  - For each pin shape, generate all valid points on the shape.



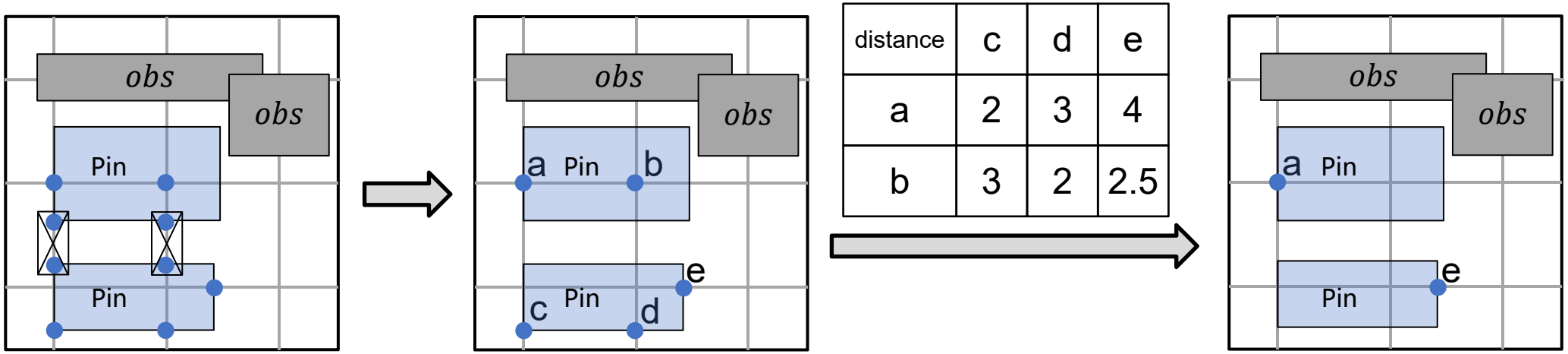
Initialize access points

Check violations with the environment

Remove all violation points

Pin Access

- Resolve via conflicts
  - Consider the conflict issues between pin shapes



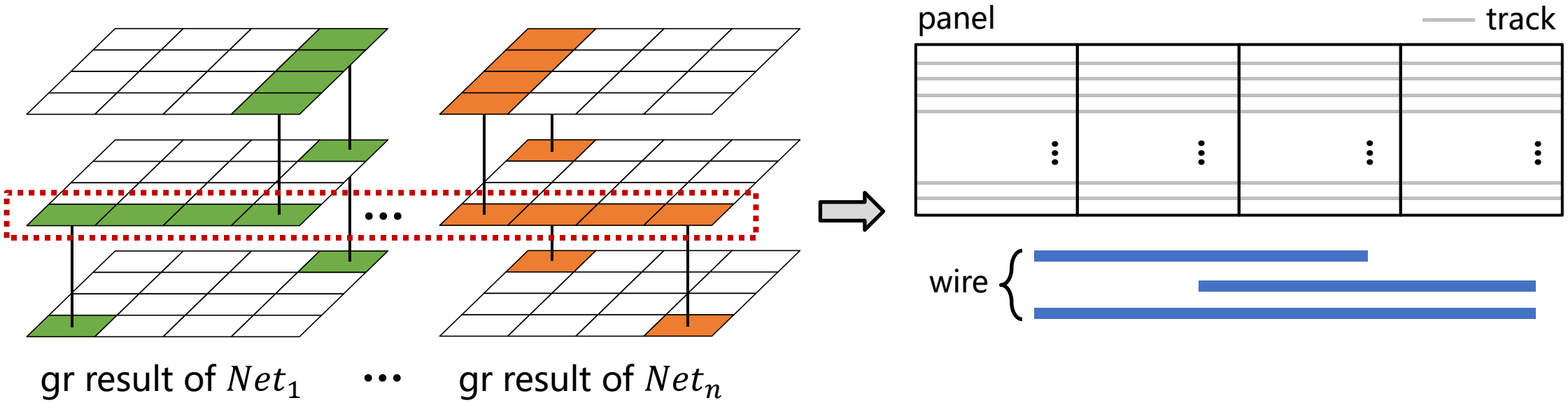
distance	c	d	e
a	2	3	4
b	3	2	2.5

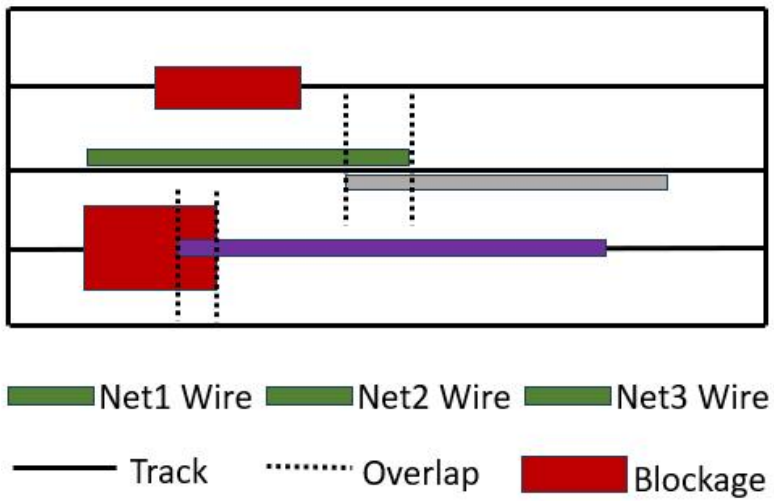
Static conflict check

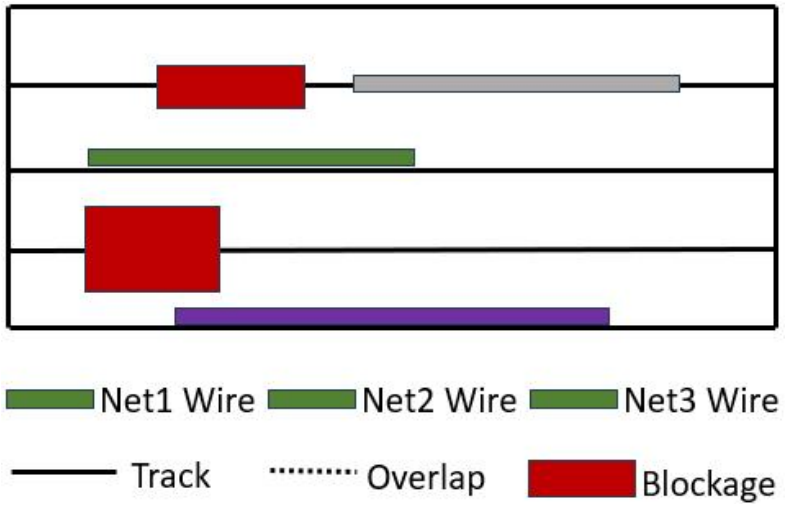
Dynamic conflict check

Track Assignment

- Modeling
  - Consider the gr results of all nets as one wire, and they will be allocated tracks within the same panel

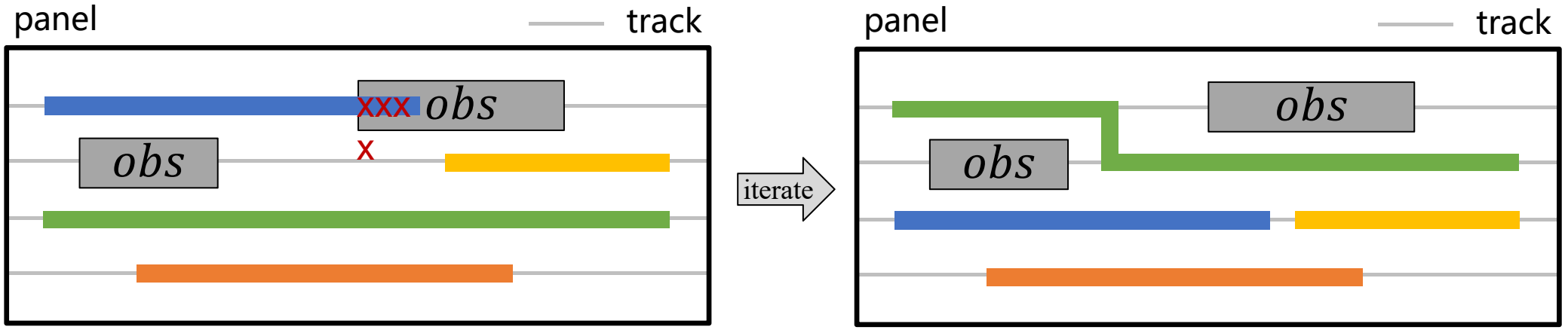



Net1 Wire
Net2 Wire
Net3 Wire
Track
Overlap
Blockage


Net1 Wire
Net2 Wire
Net3 Wire
Track
Overlap
Blockage

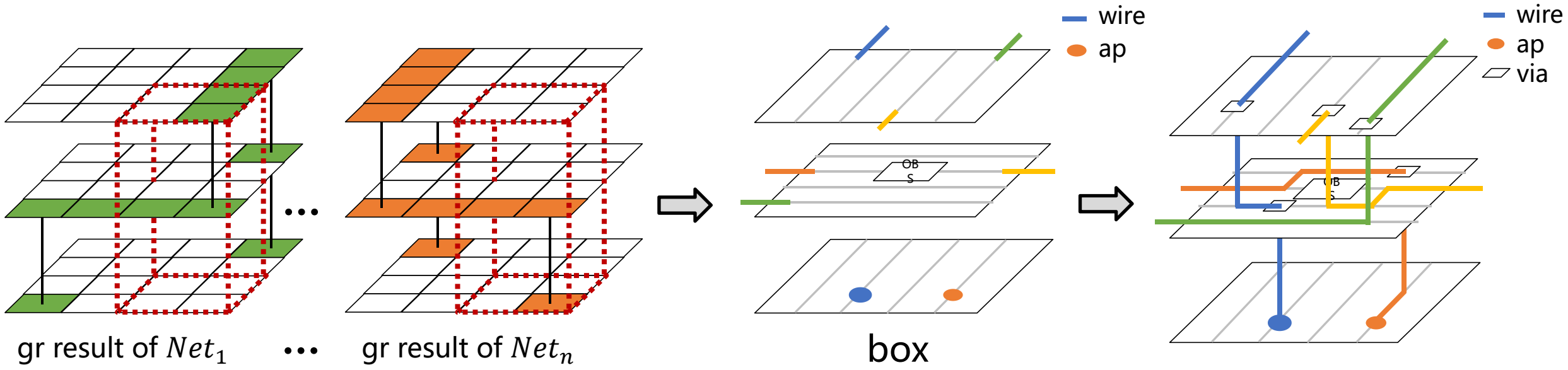
Track Assignment

- Processing
  - Sort the wires in descending order based on length
  - Obtain an initial solution using a greedy algorithm
  - Iterate simply to achieve a better solution, and accept non-prefer routing



Detailed Routing

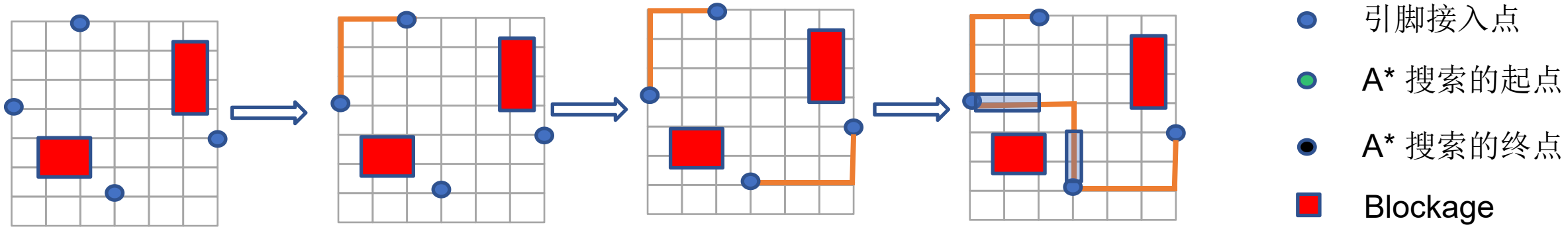
- Modeling
  - Partition the layout with Boxes, each comprised of multiple GCells
  - Inside the Box, wires(TA results) will be connected to the access points



详细布线-布线算法

- 布线算法： 传统A*路径搜索算法

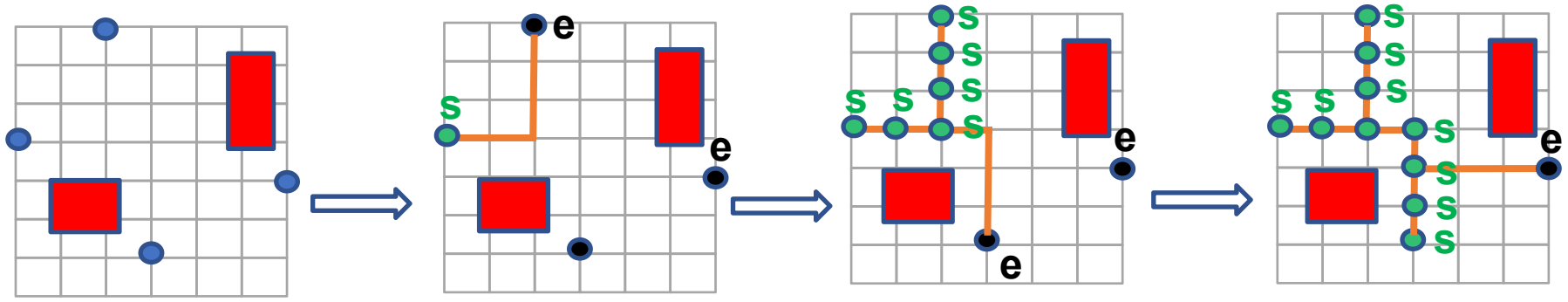
现有工作主要使用**迷宫算法**和**A*算法**进行布线，缺点是没有复用前面的布线结果，导致冗余绕线，导致线长增加，消耗额外布线资源，降低了布线质量。



详细布线-布线算法

- 多源多汇的A*路径搜索算法

针对**迷宫算法**和**A*算法**进行布线，没有复用前面的布线结果，导致冗余绕线，导致线长增加，我们改进了A*布线算法为多源多汇，避免冗余绕线。



e

Access point

A* 搜索的起点

A* 搜索的终点

Blockage

A*路径搜索cost map

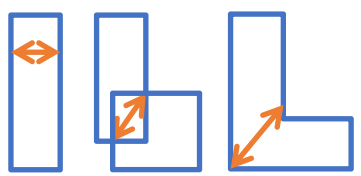
0.2	0.1	0.3	0.5
0.1	0.2	0.8	0.5
0.5	0.4	0.8	0.7
0.1	0.1	0.2	0.1

整数化

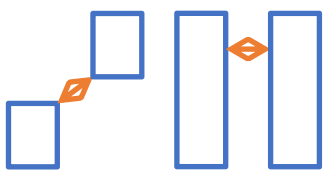
1	1	1	2
1	1	3	2
2	2	3	3
1	1	1	1

Design Rule Check (DRC)

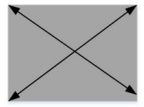
- Avoiding Design Rule Violation
  - Complex design rules
  - Cross-impact between rules
  - How to guide routing correctly



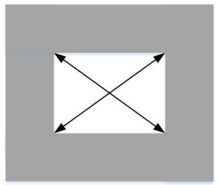
width



spacing

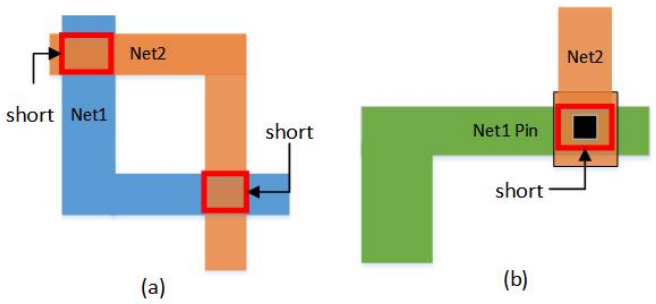


Area

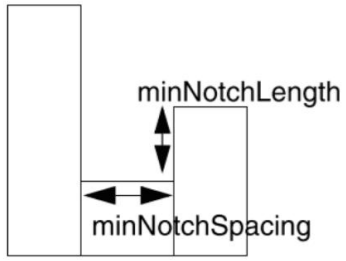


Enclosed Area

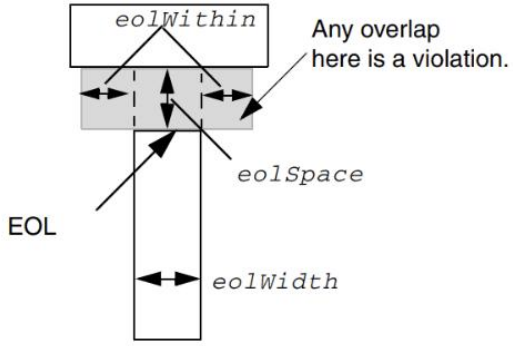
area



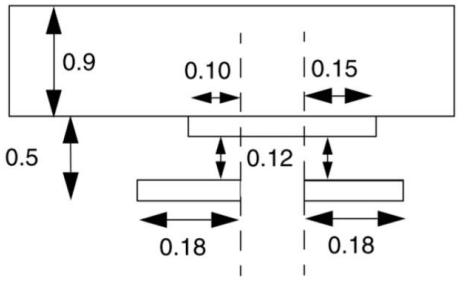
short



notch



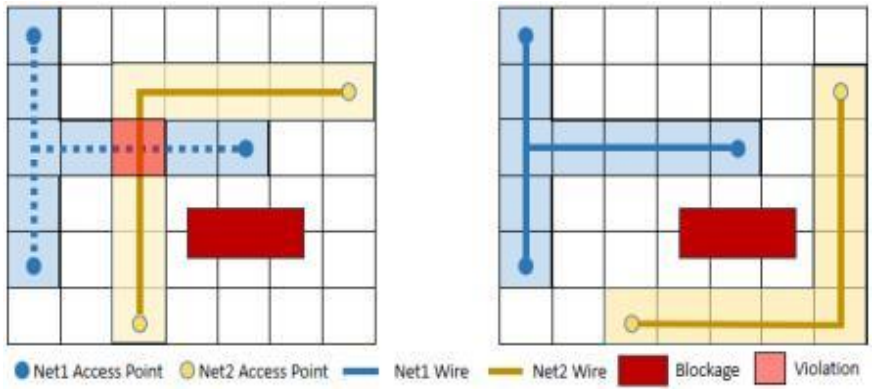
end-of-line



jog

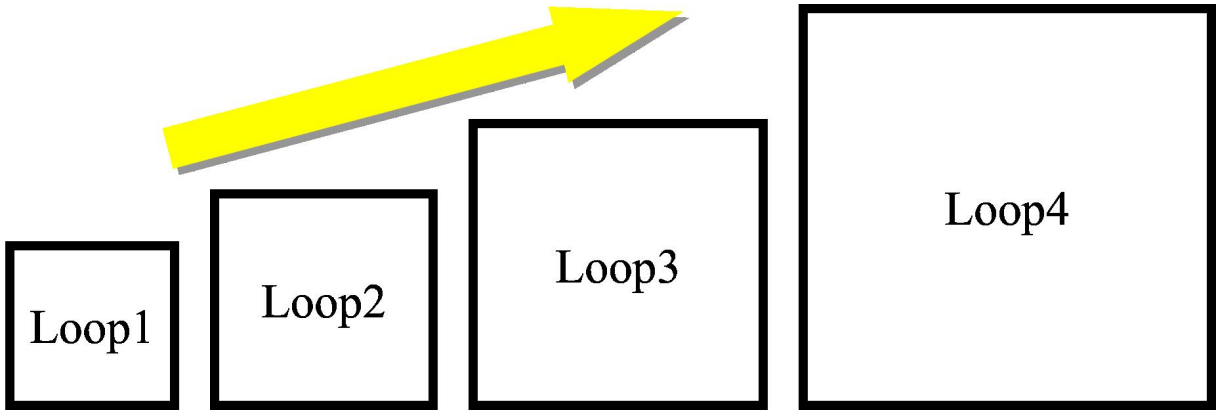
拆线重布

- 拆线重布是在布线过程中，用于解决由于不合理的布线顺序导致的绕线、资源浪费和设计规则违例（DRV）等问题的一种优化技术。布线顺序的选择会直接影响布线结果，尤其是在资源有限的情况下，不合理的布线顺序可能阻挡其他线网，导致信号连接被阻隔或者产生绕线现象。在布线过程中，某些线网会因为先前布线结果阻挡连接路径，导致布线任务无法顺利完成。例
  - 左图中，net2（黄色）先进行布线，阻断了net1（蓝色）连接点与外界的路径。
  - 当后续执行net1的布线时，由于net2的阻挡，不可避免地产生了DRV。右图则通过调整布线顺序，首先布线net1，随后布线net2。这样，net2的布线能够绕过net1的路径，避免了产生DRV。



Search & Repair

- Search&Repair fixes remaining DRC violations through multiple loops using progressively larger SBox sizes
- Note: Even if the design is DRC clean after S&R, you must still run a sign-off DRC checker (Caliber).
  - Routing DRC rules are a subset of the complete technology DRC rules
  - IC Compiler works on the FRAM view, not the detailed transistor-level (CEL) view



iEDA

01 Introduction

02 Routing Techniques

03 Global Routing

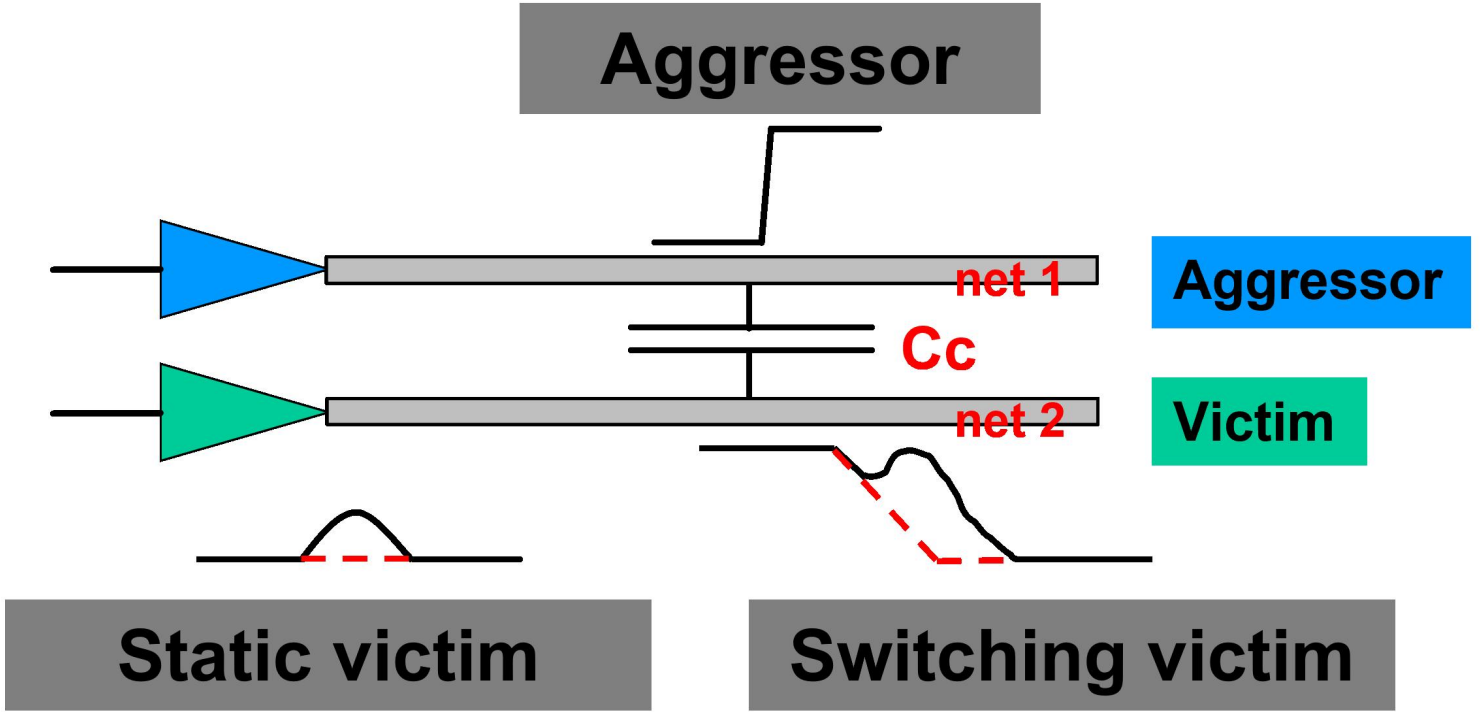
04 Detail Routing

05 Beyond Routing

What is Crosstalk?

iEDA

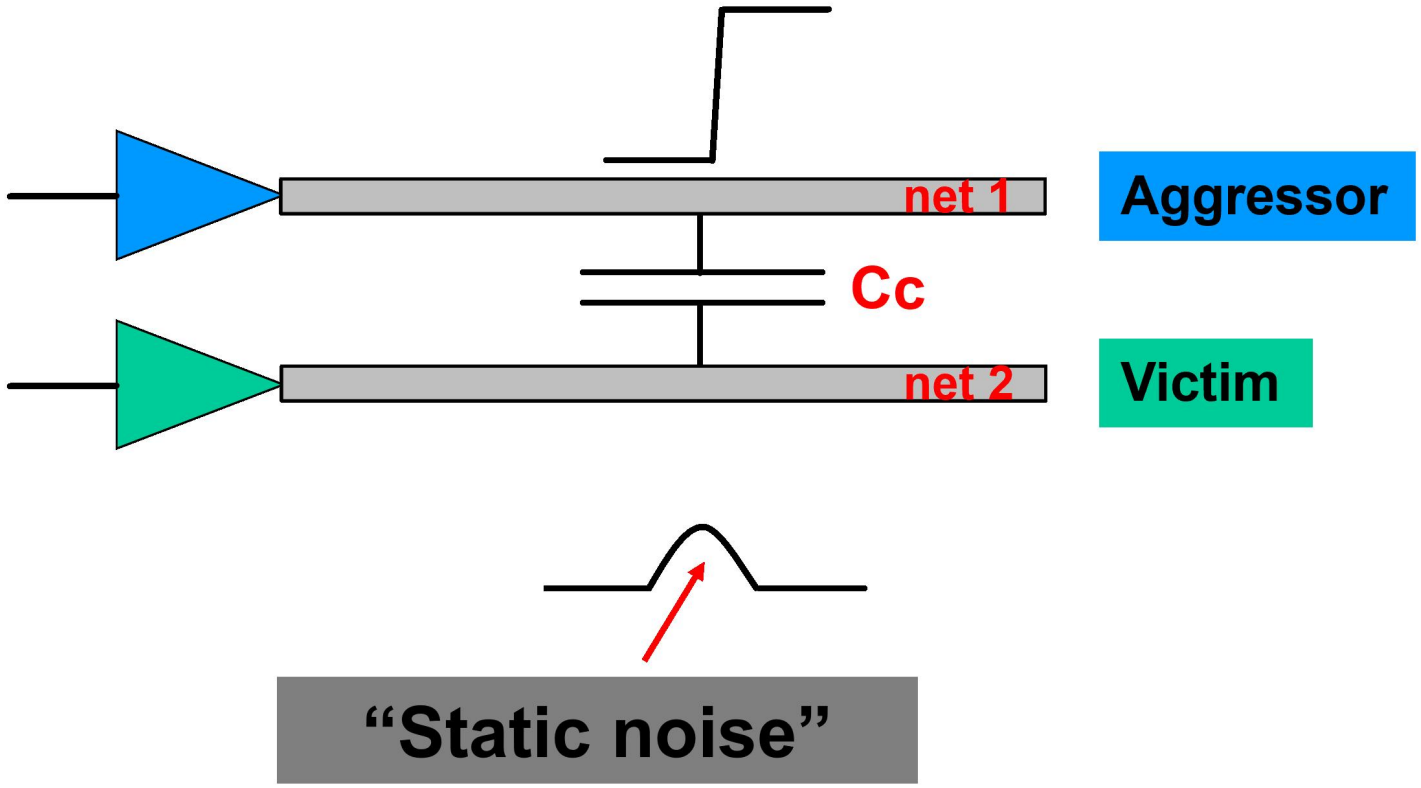
Crosstalk is the transfer of a voltage transition from one switching net (aggressor) to another static or switching net (victim) through a coupling capacitance (Cc)



Crosstalk-Induced Noise (aka Glitches)

iEDA

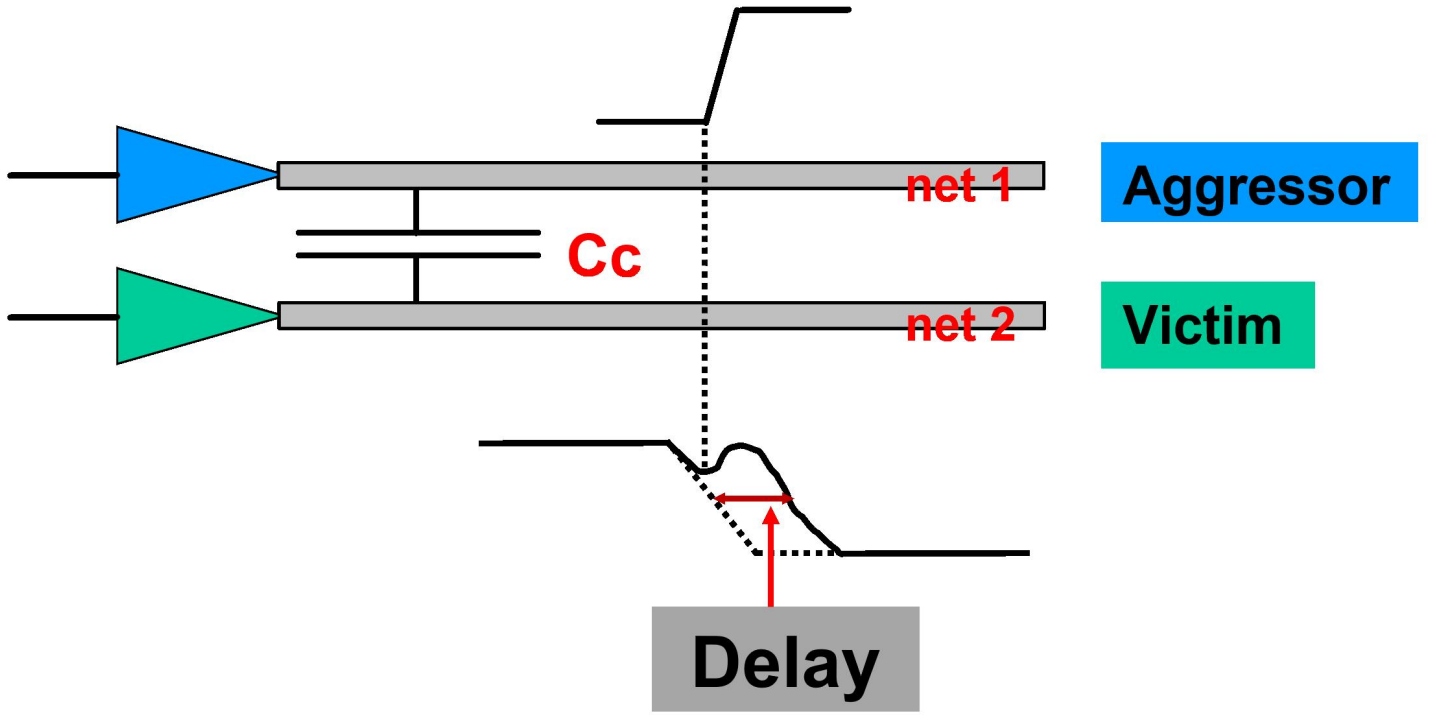
Aggressor nets can create crosstalk-induced noise on static victim nets, also called as “static noise”



Crosstalk-Induced Delay

iEDA

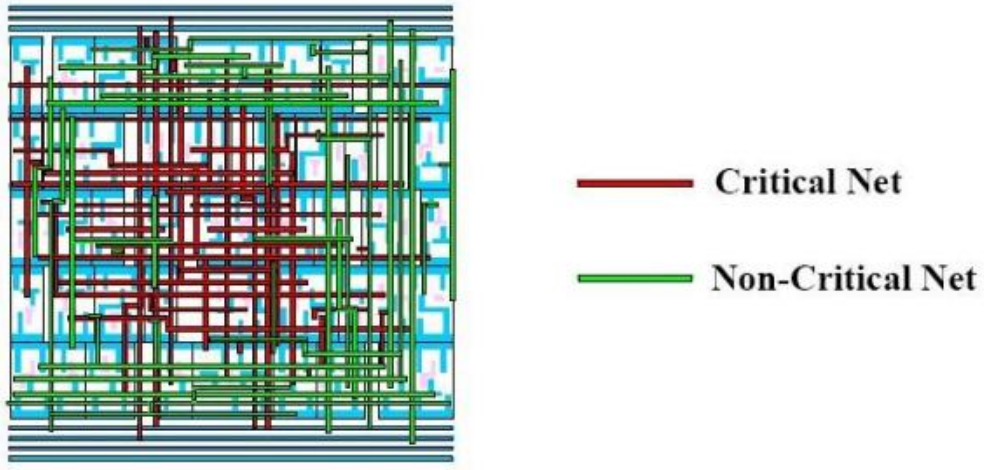
Aggressor/victim nets with overlapping timing windows can cause “crosstalk-induced delay” on victim nets.
This can lead to a speed-up or a slow-down of the victim net.



Timing-Driven Routing

iEDA

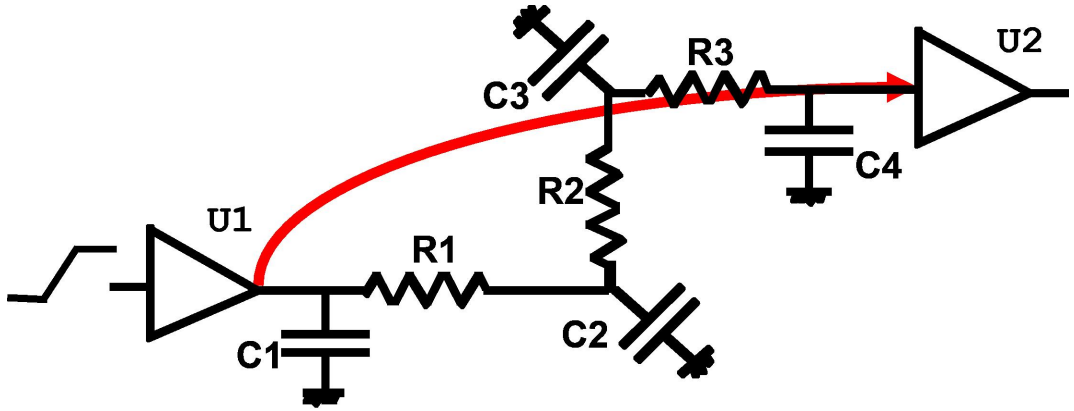
- Routing along the timing-critical path is given priority:
  - Creates shorter, faster connections
- Non-critical paths are routed around critical areas:
  - Reduces routability problems for critical paths
  - Does not adversely impact timing of non-critical paths



Calculating Cell and Net Delay

iEDA

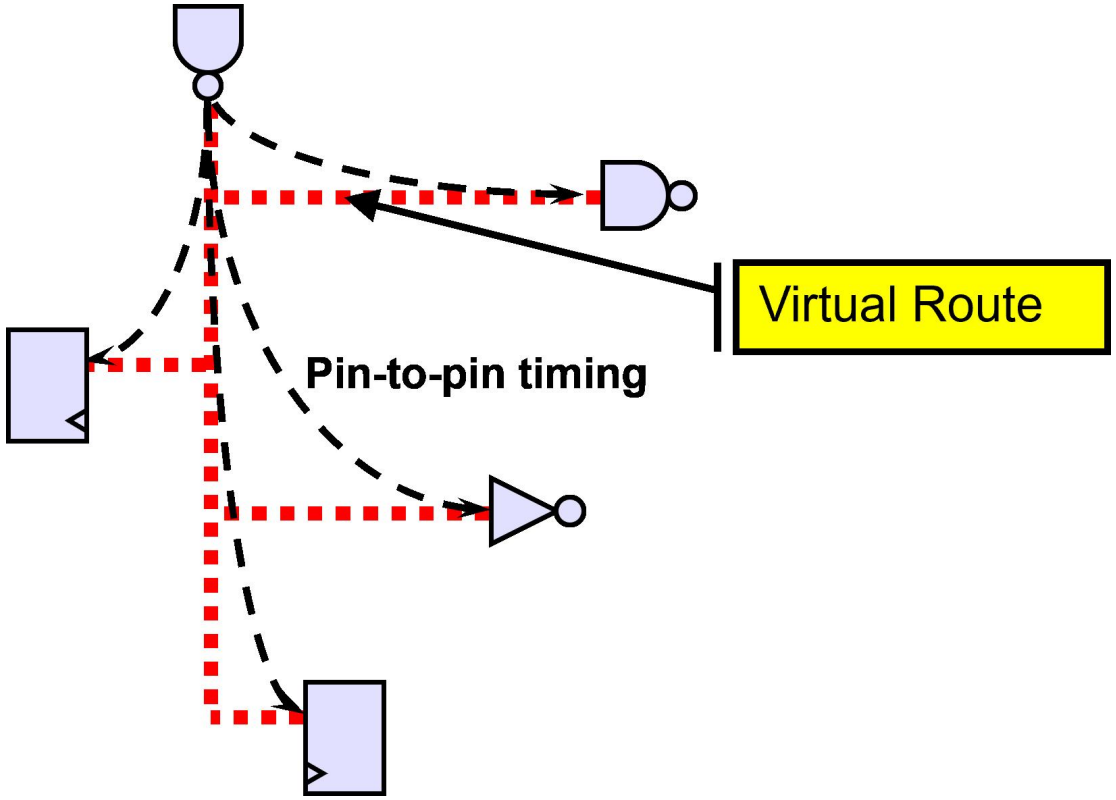
- Now that R and C are known from TLU+, the delays can be calculated
- For Cell Delays, only Ctotal / Ceff is needed
- Calculating Net Delay is done using Delay Calculation algorithms: Elmore, Arnoldi



PreRoute Delay Calculation Algorithm

iEDA

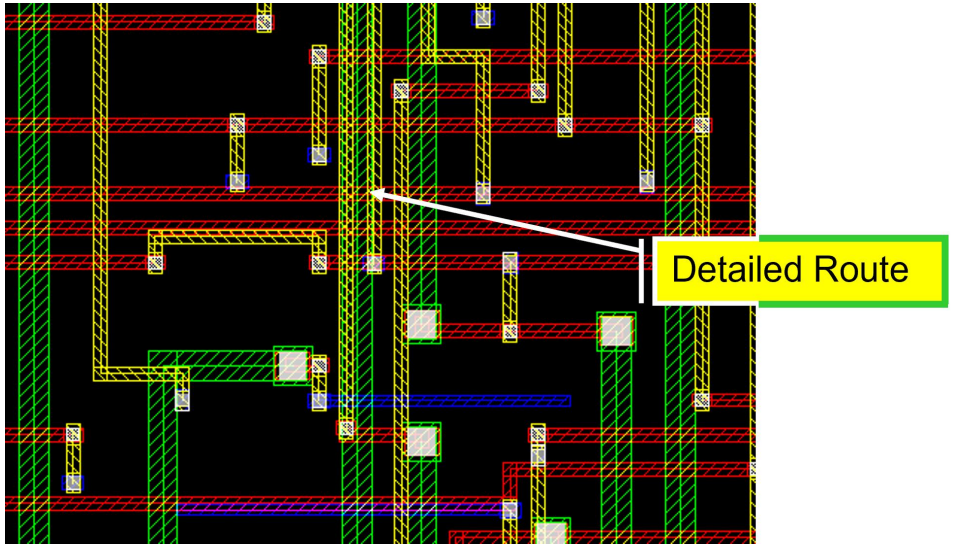
- Prior to routing, net geometry is estimated based on a Virtual Route
- Since Virtual Routing is only an estimate,an Elmore model is used for delay calculation



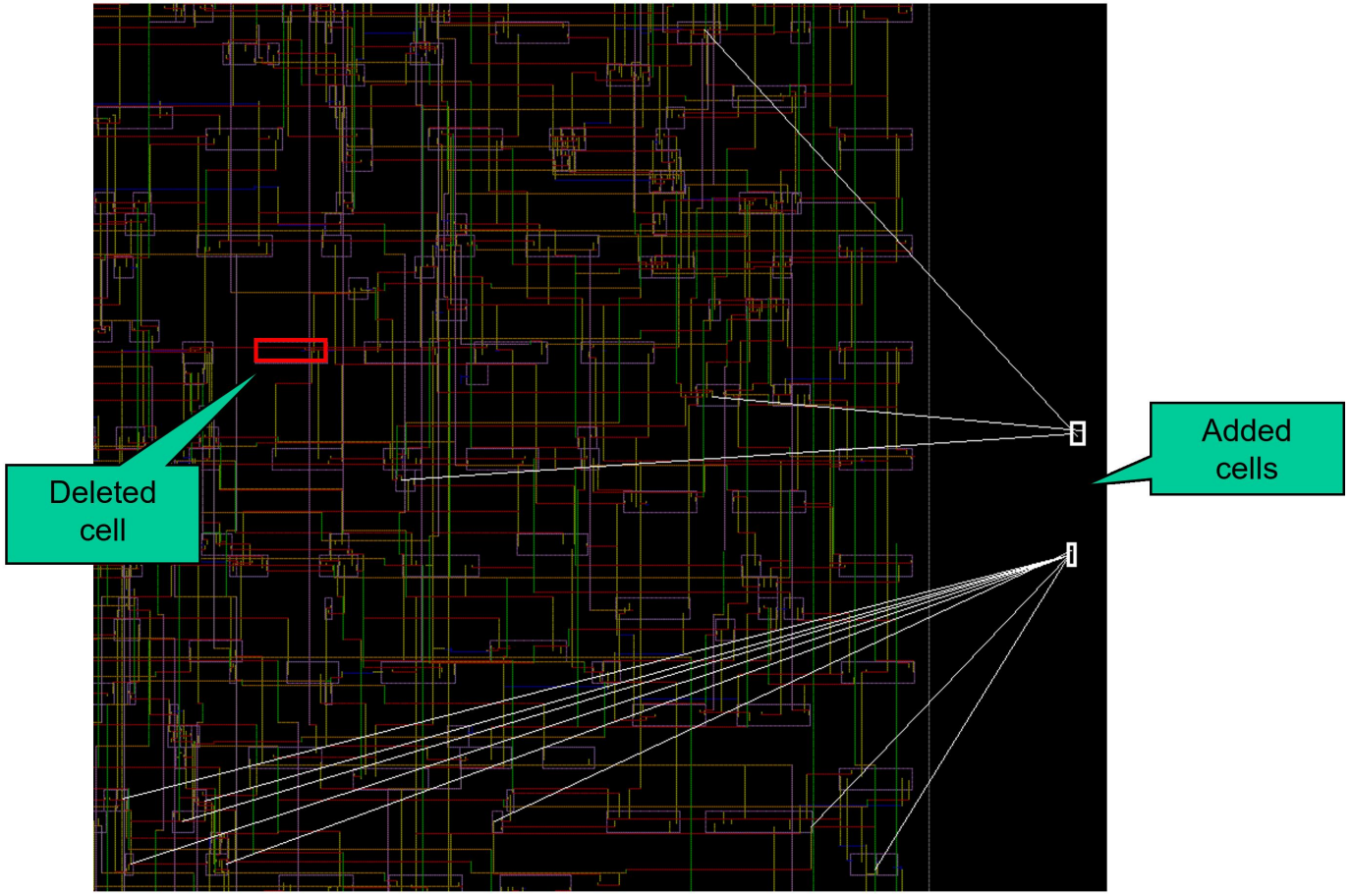
PostRoute Delay Calculation Algorithms

iEDA

- After routing, detailed nets are available and extractioncan be more accurate
- By default, Elmore is still used
- Arnoldi can be turned on for postroute calculations

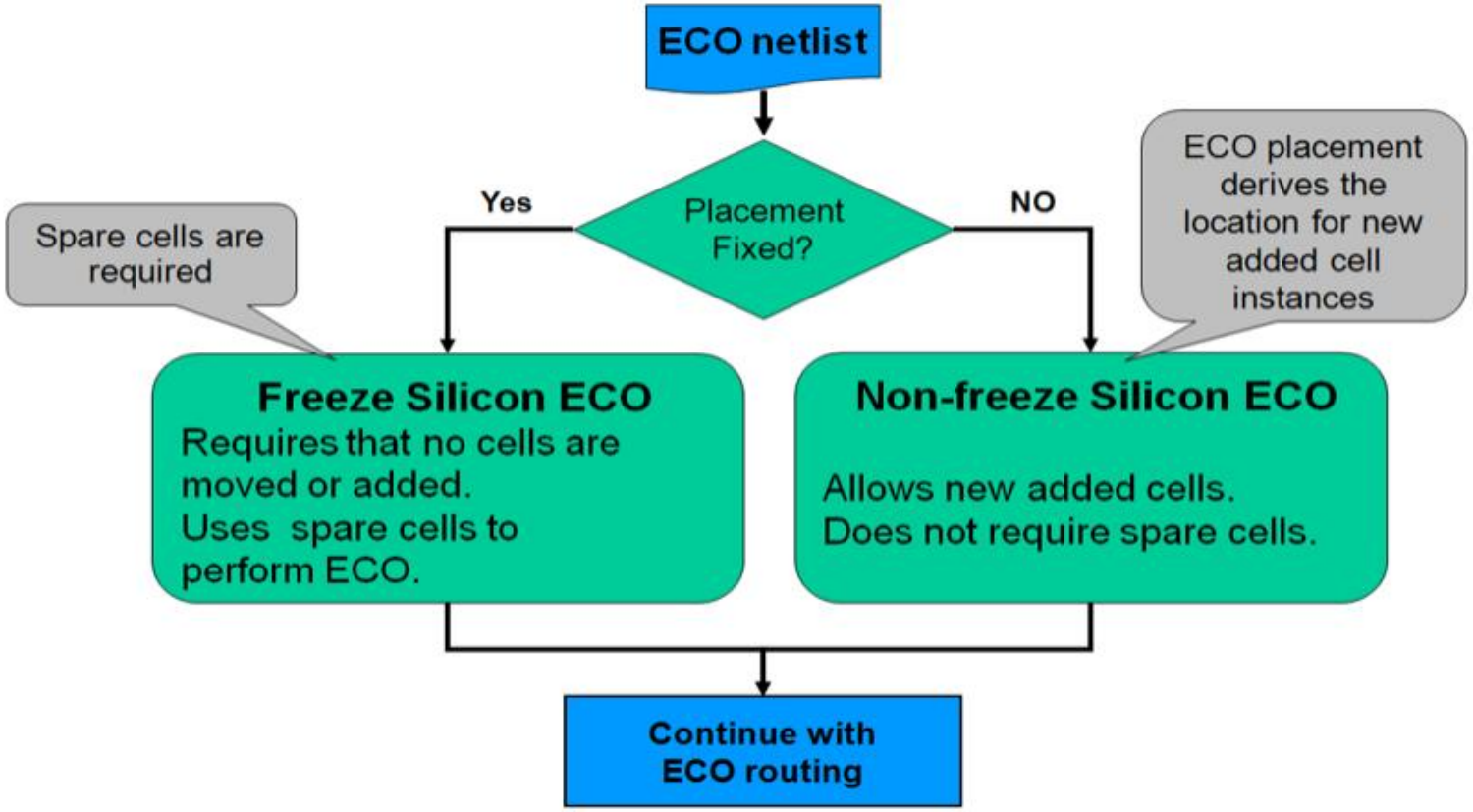


Engineering change order



Functional changes occur late in the design cycle

Two Types of ECO Flows



Functional ECO Flows

iEDA

- Non-Freeze silicon ECO
  - Pre-tapeout, no restriction on placement or routing
  - Minimal disturbances to the existing layout
  - ECO cells are placed close to their optimal locations
- Freeze silicon ECO
  - Post-tapeout, metal masks change only using previously inserted spare cells
  - Cell placement remains unchanged
  - ECO cells are mapped to spare cells that are closest to the optimal location
  - Deleted cells become spare cells

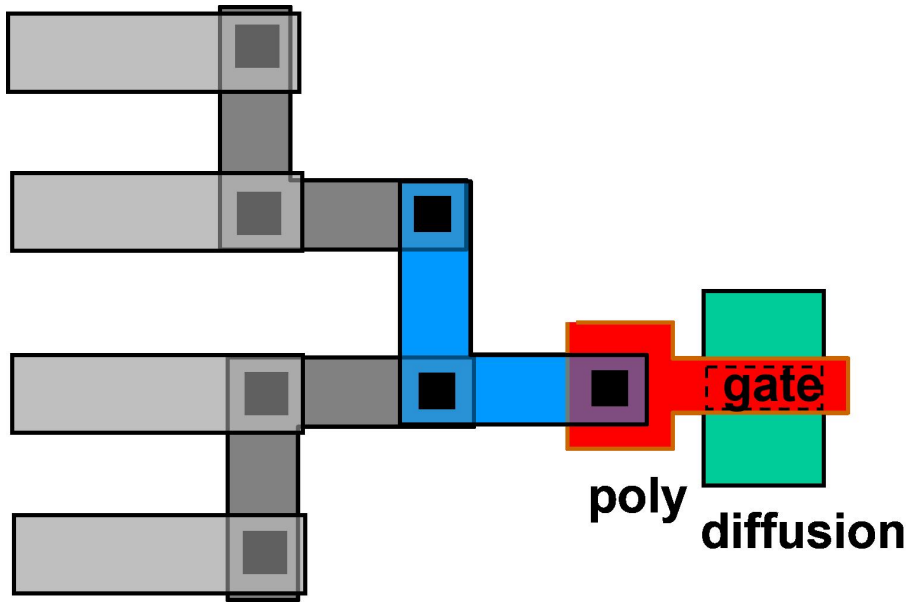
Chip Finishing Flow

iEDA

- PnR Compiler can address several issues to increase manufacturing yield:
  - Gate Oxide integrity → antenna fixing
  - Via resistance and reliability → extra contacts
  - Random Particle defect → Wire spreading
  - Metal erosion → metal slotting
  - Metal liftoff → metal slotting
  - Metal Over-Etching → metal fill

Antenna Rules

- As length of wire increases during processing, the voltage stressing the gate oxide increases
- Antenna rules define acceptable length of wires



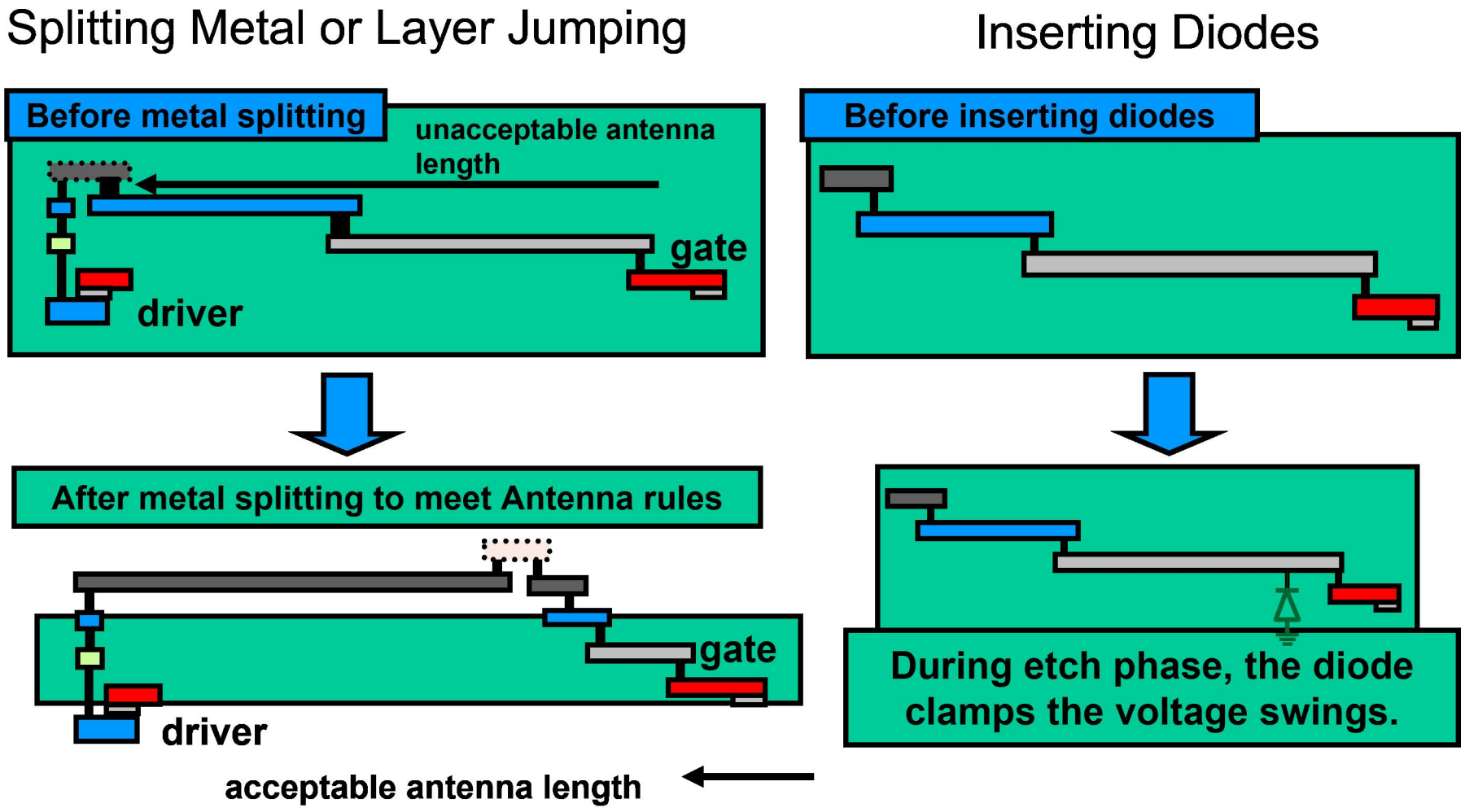
Antenna Ratios:

$$\frac{\text{Area of Metal Connected to Gate}}{\text{Combined Area of Gate}}$$

Or

$$\frac{\text{Area of Metal Connected to Gate}}{\text{Combined Perimeter of Gate}}$$

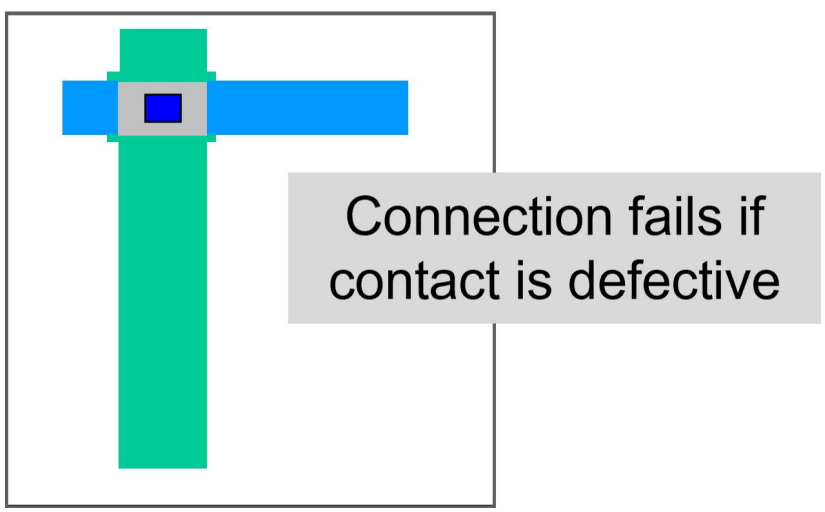
Fixing Antenna Problems

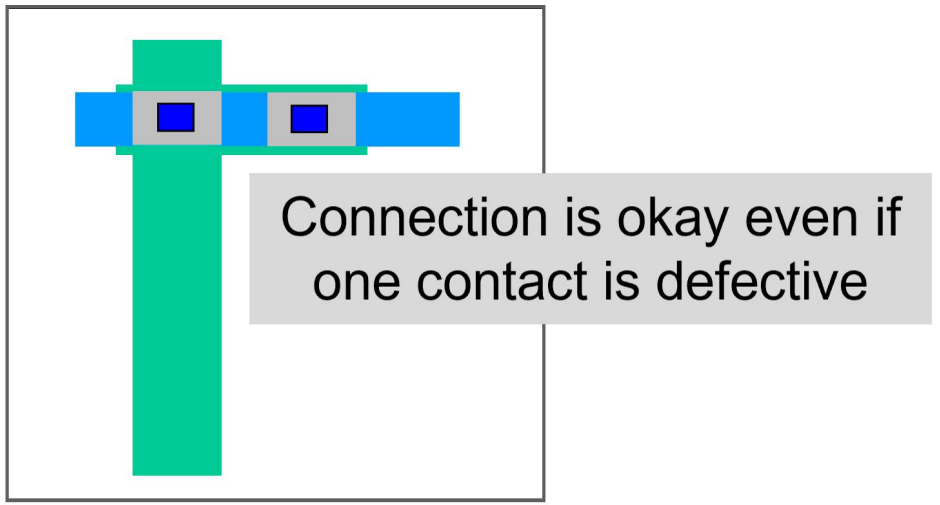


Voids in Vias During Manufacturing

iEDA

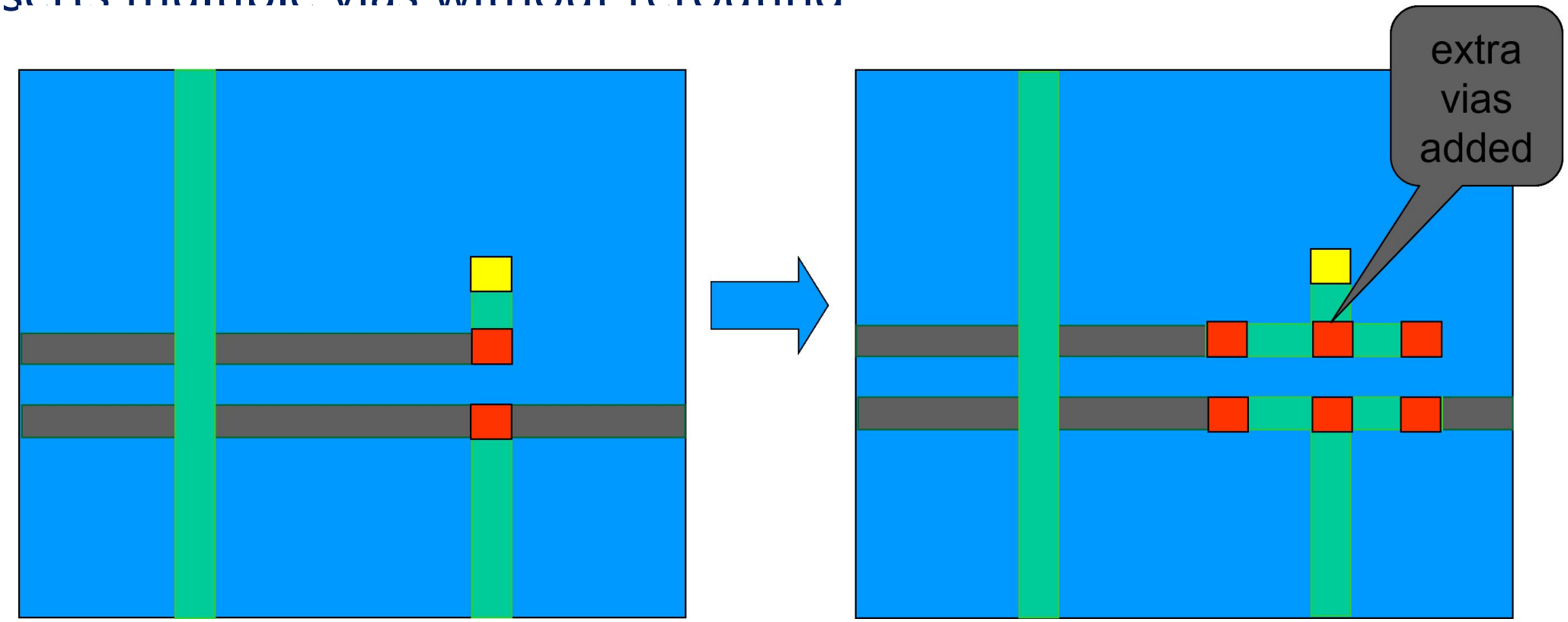
- Voids in vias is a serious issue in manufacturing
- Two solutions are available:
- Reduce via count: via optimization techniques are employed in
- routing stage
- Add backup vias: known as redundant vias





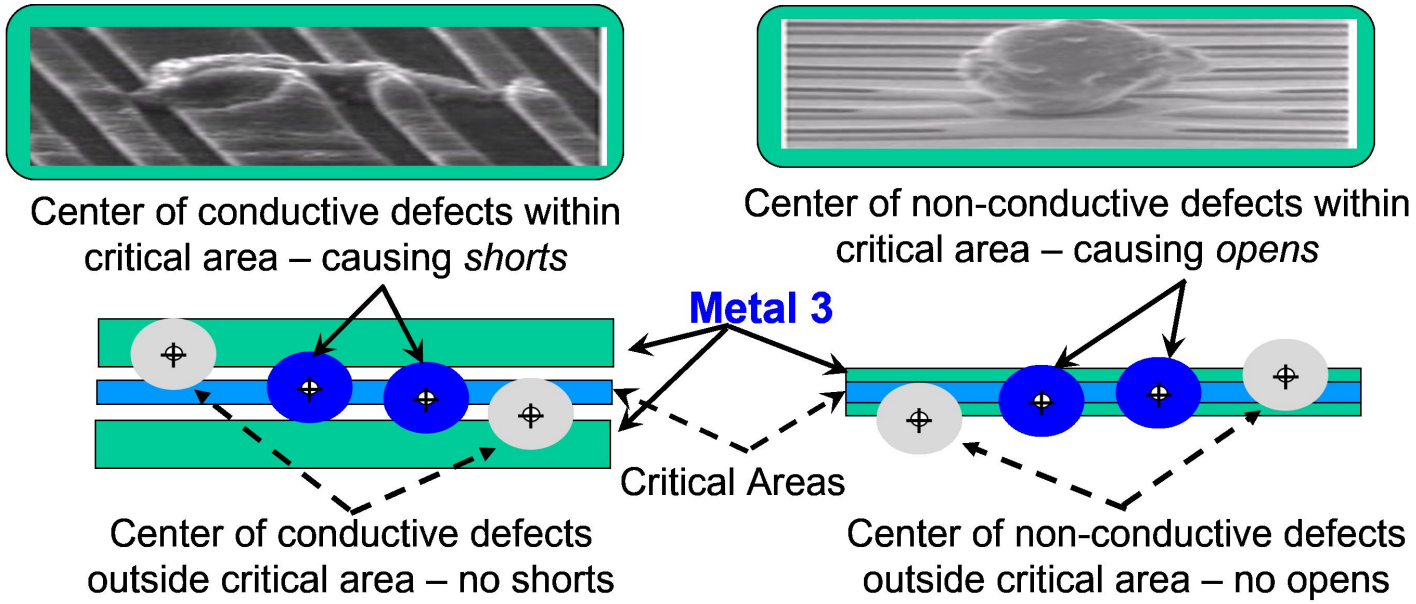
Via Resistance and Reliability

- Replacing one via with multiple vias can improve yield &
- timing (series R reduction)
- Inserts multiple vias without rerouting



Wire Spreading: Random Particle Defects

- Random missing or extra material causes opens or shorts during the fabrication process
- Wires at minimum spacing are most susceptible to shorts
- Minimum-width wires are most susceptible to opens



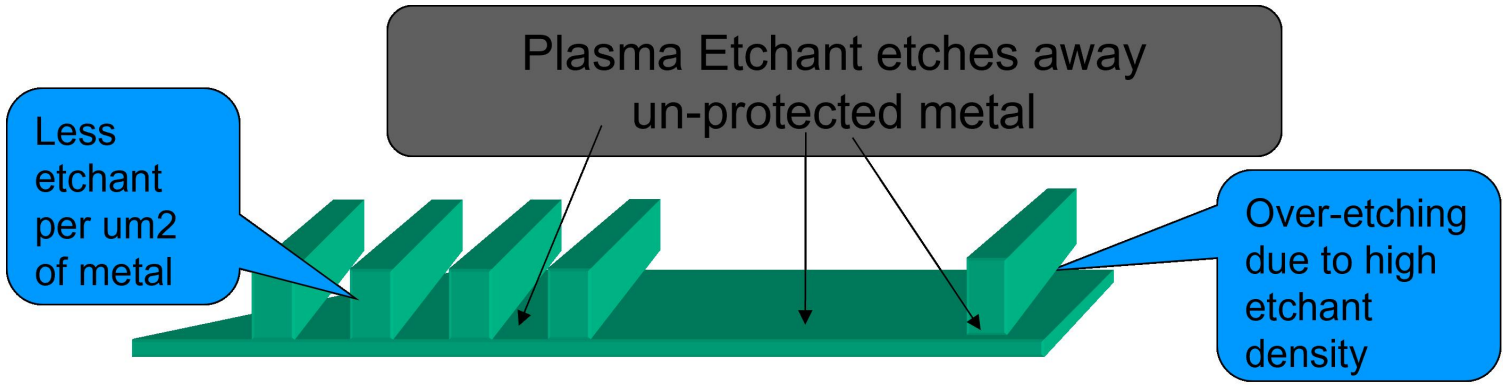
Filler Cell Insertion

iEDA

- Metal fill insertion helps
  - To fill layers of choice including poly
  - To insert floating vias
  - To do area based metal fill
  - To specify required width (timing driven metal fill doesn’ t fill wire tracks around critical nets)
- For better yield, density of the chip needs to be uniform
- Some placement sites remain empty on some rows

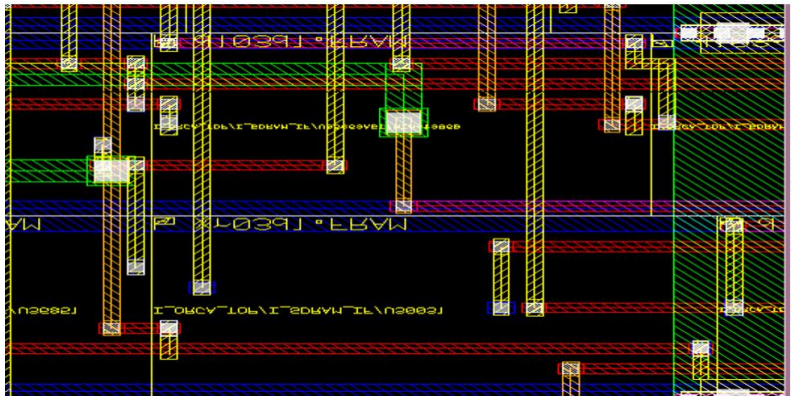
Problem: Metal Over-Etching

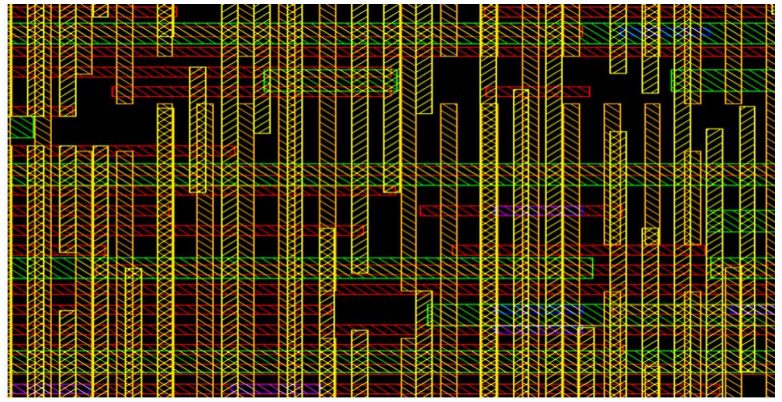
- A narrow metal wire separated from other metal receives a higher density of etchant than closely spaced wires
- The narrow metal can get over-etched
- Minimum metal density rules are used to control this
- Too much etchant in contact with too little metal → over-etched metal



Solution: Metal Fill

- Fills empty tracks with metal shapes to meet the minimum metal density rules
- Uses up most of the remaining routing resource:
  - No further routing or antenna fixes can be done

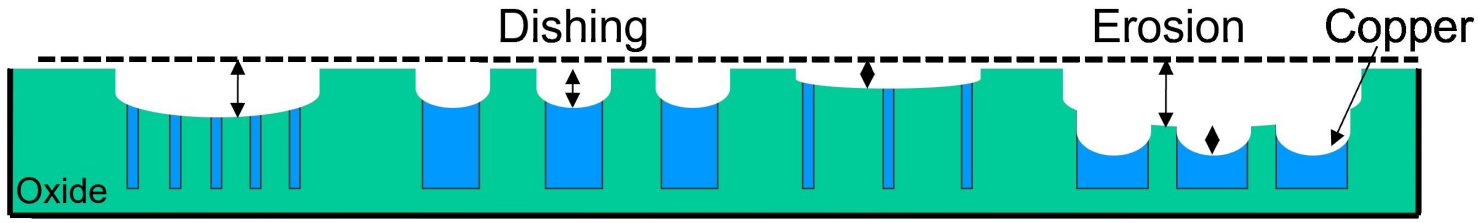




- Metal filling is done to improve process planarization, which is important for processes with a large number of metal layers.

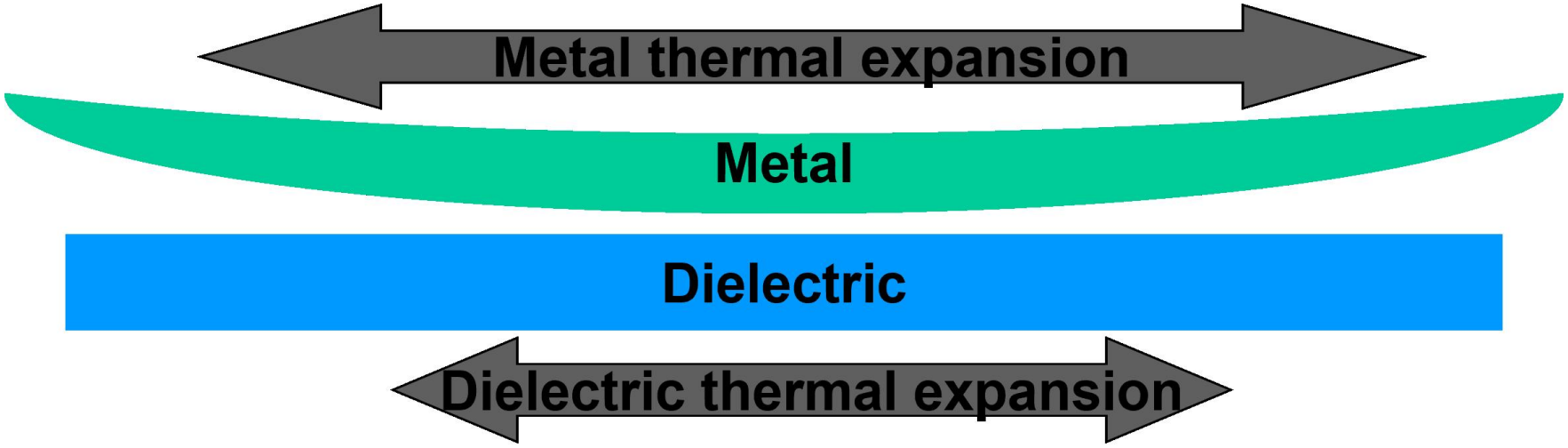
Problem: Metal Erosion

- The wafer is made flat (planarized) by a process called Chemical Mechanical Polishing (CMP)
- Metals are mechanically softer than dielectrics:
  - CMP leaves metal tops with a concave shape - dishing
  - The wider the metal the more pronounced the dishing
  - Wide traces with little intervening dielectric and can become
- quite thin – dishing this severe is called erosion Process rules specify maximum metal density per layer to minimize erosion



Problem: Metal Liftof

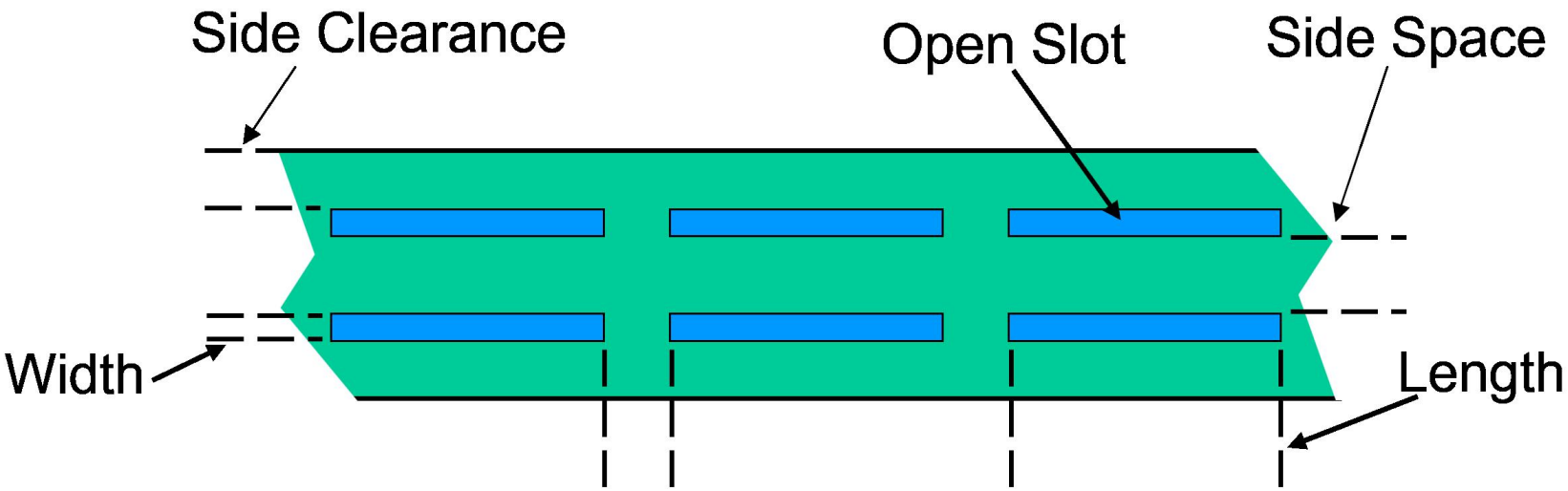
- Conductors and Dielectrics have different coefficients of thermal expansion:
  - Stress builds up with temperature cycling
  - Metals can delaminate (lift off) with time
  - Wide metal traces are more vulnerable than narrow ones
- Maximum metal density rules also address this issue



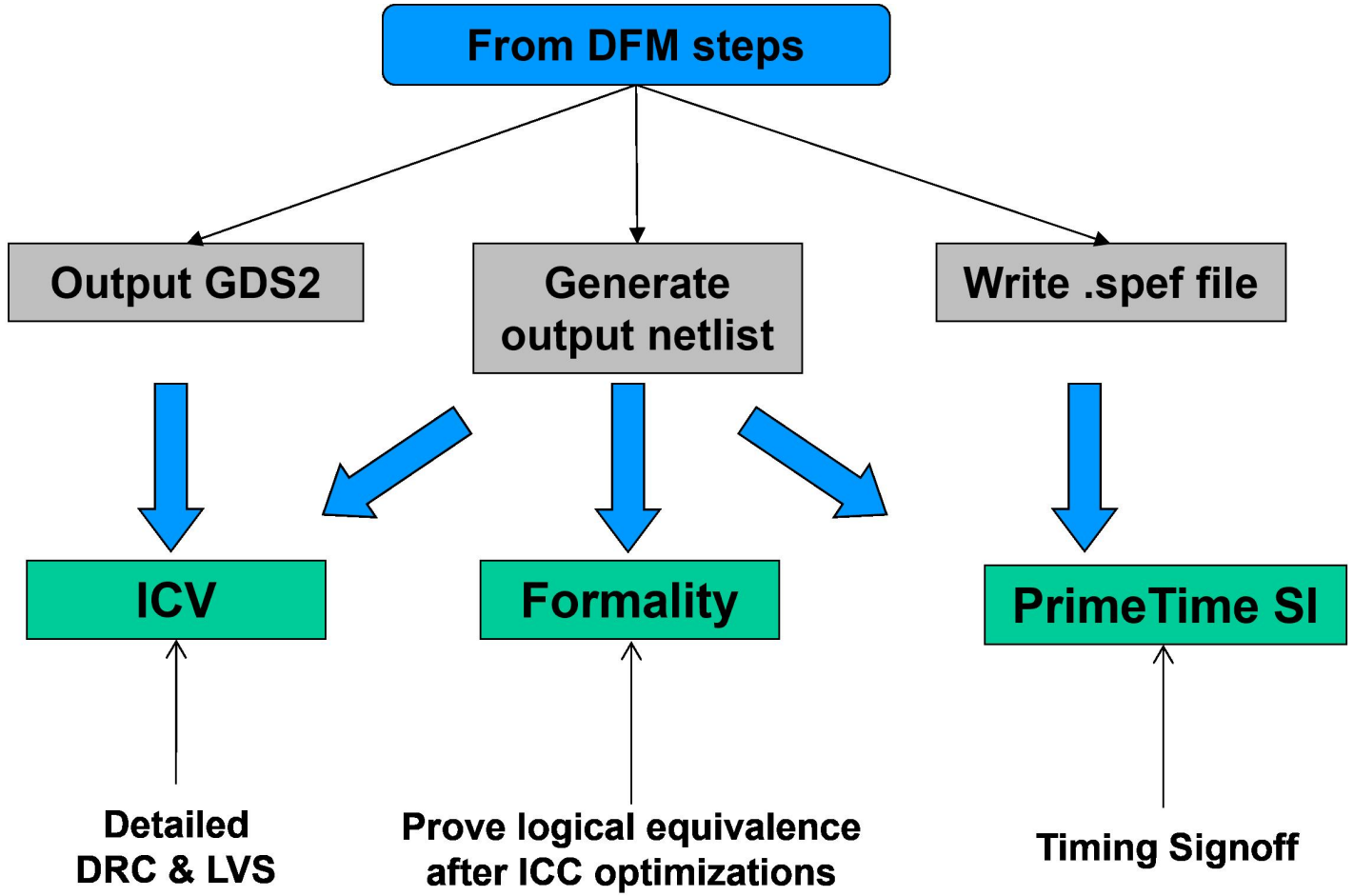
Solution: Metal Slotting

iEDA

- Slotting wide wires reduces the metal density
- Slots minimize stress buildup, reducing liftoff tendency
- Primarily used on Power and Ground traces:
  - Can apply to any other net if wide enough
- Slotting parameters can be set layer by layer



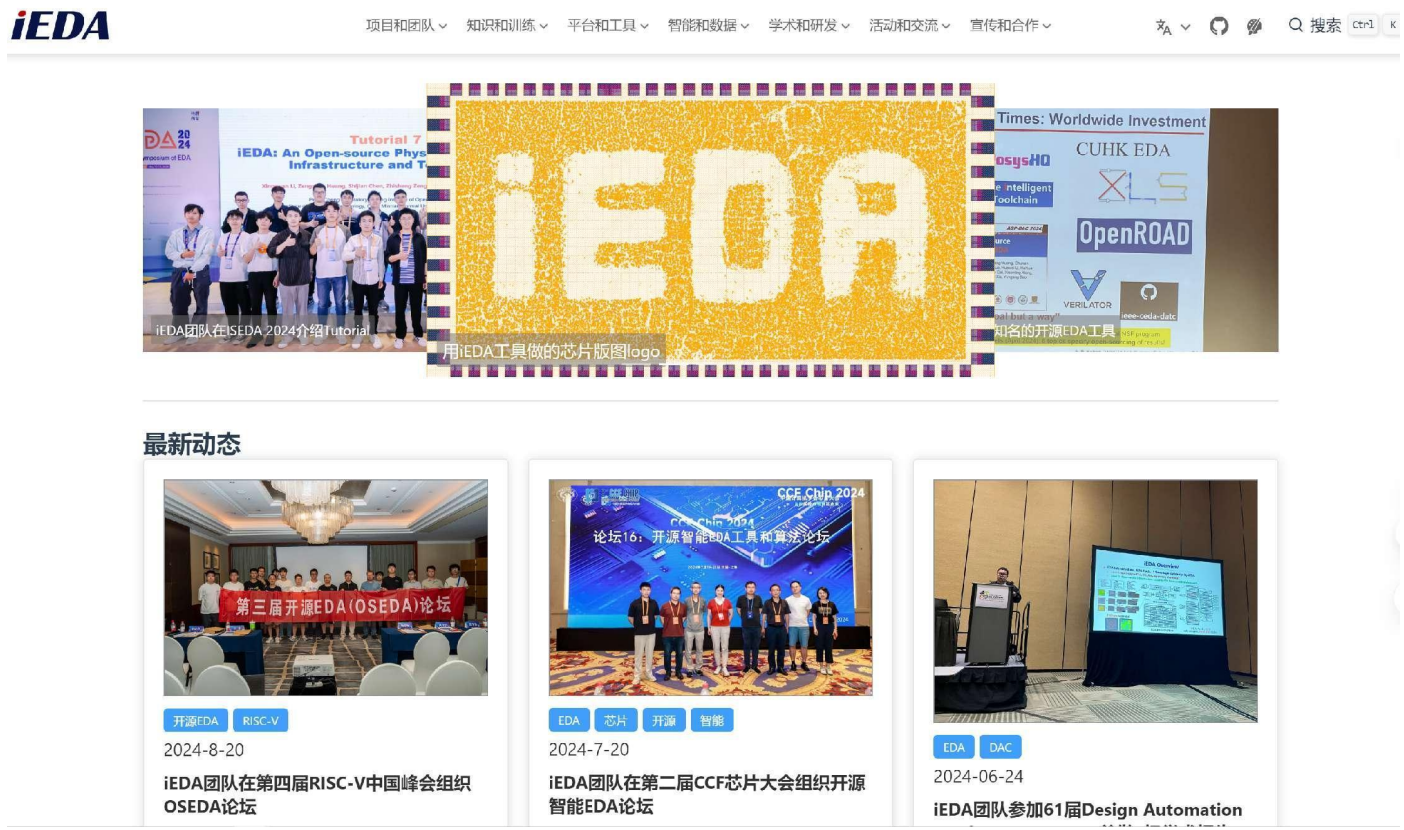
Final Validation



Conclusions

iEDA

- Clock Tree Problem
- Clock Concepts
  - Objectives：Skew, Latency, Power, Wirelength, Buffer Area
  - Constraints：Fanout, Slew, Load,
- Clock Topology
  - Tree, Grid, Spine, Hybrid
- Clock Routing
  - H-tree, GH-tree, RMST, SALT, ZST/BST/UST，CBS
- Clock Optimization
  - Buffer Insertion, Wire Sizing, Gate Sizing, Wire Snake, Wire Linking, Clock Gate
  - Clock and Data Concurrent Optimation (CCOPT)



Thanks

iEDA website: ieda.oscc.cc

李兴权（Xingquan Li）
fzulxq@gmail.com