

algorithm

n+e

July 2017

1 Algorithm

Input: 当前芯片的布线状态，其中靠近原点的子区域尚未布线

Output: 在该子区域布线之后的芯片布线状态

```
1 while 子区域中连线起点尚未全部连接  $\wedge$  有剩余连线终点 do
2   if 已经到达芯片角落 then
3     | 退出该循环;
4   end
5   if 连线终点落后于连线起点 then
6     | 选择  $n_x$  ( $n_y$ ) 所在的那一列 (行), 左一个右一个地连出去,
        | 直到边界被连接的连线终点连续排列;
7   end
8   Call function Link_in to obtain  $pt_{x1}, pt_{y1}$ , 以及备选的两条线路;
9   计算  $f_x, f_y$  的值, 如果某个值为 1 则置该侧线路长度为  $+\infty$ ;
10  if 某侧的  $f$  值不为 1  $\wedge$  计算该侧线路失败 then
11    | return 该子区域无合法布线状态;
12  end
13  比较两条线路长度, 将长度小的一条记录至布线状态中;
14  if 另一条线路的起点如果连向对侧的长度比该线路要长 then
15    | 将该线路也记录至布线状态中;
16  end
17 end
18 if 子区域中连线起点尚未全部连接  $\wedge$  无剩余连线终点 then
19   | return 该子区域无合法布线状态;
20 end
21 while 子区域中连线起点尚未全部连接 do
22   if 选取某一列全部连接至  $x$  轴的总长度要比选取某一行全部连
        | 接至  $y$  轴的总长度要短 then
23     | 选取  $n_x$  所在的一列, 将其全部连接至  $x$  轴;
24   else
25     | 选取  $n_y$  所在的一行, 将其全部连接至  $y$  轴;
26   end
27 end
28 return 当前芯片的布线状态;
```

Algorithm 1: 子区域中的布线算法.

Input: The length $node_x$ and width $node_y$ of the biochip.

Output: 最小的间隔 d 、布线总长度和布线方案

```

1 if  $\min\{N, M\} \leq 2$  then
2   | 直接构造出方案并返回;
3 end
4 将连线起点划分成 4 个子区域;
5 确定  $d$  的上下界, 使用二分法判定在该状态下是否有可行方案;
6 foreach 二分出来的  $d$  do
7   | Calculating total length and width of biochip;
8   | Connecting origins near central axis;
9   for 每个子区域 do
10    | Call Algorithm 1 to obtain 子区域中的布线方案 with 当前芯
11    | 片布线状态 as input;
12    | if 该函数返回布线失败 then
13    |   |  $d$  不合法, 退出该循环;
14    |   end
15    | 绕坐标轴进行坐标变换;
16  end
17 end
18 return 最小的间隔  $d$ 、布线总长度和布线方案;

```

Algorithm 2: Complete routing method of RRBI.????