

# 面向超大规模集成电路物理设计的通孔感知的并行层分配算法

刘耿耿<sup>1</sup>, 李泽鹏<sup>1</sup>, 郭文忠<sup>1</sup>, 陈国龙<sup>1</sup>, 徐 宁<sup>2</sup>

(1. 福州大学计算机与大数据学院, 福建福州 350116; 2. 武汉理工大学信息工程学院, 湖北武汉 430070)

**摘 要:** 随着集成电路规模的日益增长, 需要处理的线网数量显著增多, 层分配算法运行时间增大成为限制高效设计布线方案的重要因素; 此外在生产工艺中, 通孔的制造成本较高. 针对以上两个问题, 本文提出了两种新颖的策略分别用于优化算法运行时间和通孔数量: (1)一种高效的基于区域划分的并行策略, 实现各区域在并行布线阶段负载均衡, 以提高并行布线的效率; (2)基于线网等效布线方案感知的通孔优化策略, 决定各线网对布线资源使用的优先级, 进而减少层分配方案的通孔数量. 最终将上述两种策略相结合, 提出了一种面向超大规模集成电路物理设计的通孔感知的并行层分配算法. 实验结果表明该算法对通孔数量和运行时间均有良好的优化效果.

**关键词:** 并行算法; 层分配; 通孔; 区域划分; 负载均衡; 超大规模集成电路

**中图分类号:** TP391.7

**文献标识码:** A

**文章编号:** 0372-2112(2022)11-2575-09

**电子学报 URL:** <http://www.ejournal.org.cn>

**DOI:** 10.12263/DZXB.20211065

## Via-Aware Parallel Layer Assignment Algorithm for VLSI Physical Design

LIU Geng-geng<sup>1</sup>, LI Ze-peng<sup>1</sup>, GUO Wen-zhong<sup>1</sup>, CHEN Guo-long<sup>1</sup>, XU Ning<sup>2</sup>

(1. College of Computer and Data Science, Fuzhou University, Fuzhou, Fujian 350116, China;

2. School of Information Engineering, Wuhan University of Technology, Wuhan, Hubei 430070, China)

**Abstract:** As the scale of integrated circuits grows, the number of nets greatly increases, which makes the runtime of layer assignment algorithm increase and becomes an important limiting factor of efficient routing algorithm. Besides, in the manufacture, vias always take high cost. Accordingly, this paper presents two strategies to reduce runtime and the number of vias: (1) an efficient region-division based parallel strategy, which realizes load balancing of parallel routing to improve the efficiency of routing algorithm; (2) an equivalent routing solution aware via optimization strategy, which determines the priority of each net in using routing resource to reduce the number of vias of layer assignment. Furthermore, combining the above two strategies, this paper proposes a via-aware parallel layer assignment algorithm for very large scale integration (VLSI) physical design. The experimental results show that the proposed algorithm is able to optimize the number of vias significantly and reduce runtime simultaneously.

**Key words:** parallel algorithm; layer assignment; via; region-division; load balancing; VLSI

## 1 引言

在超大规模集成电路的物理设计流程中, 层分配是十分重要的阶段. 层分配需要使用通孔来连接位于不同布线层上的导线. 而通孔的使用数量是影响制造成本的重要因素, 减少通孔的数量有利于节约制造成本. 此外, 随着集成电路的规模日益增长, 需要处理的线网数量显著增多, 层分配算法的运行时间增大成为

限制高效设计布线方案的重要因素. 同时, 随着多核处理器的发展, 并行算法能够更充分地发挥现代处理器的计算能力来处理超大规模集成电路的布线问题.

目前, 层分配作为布线的关键阶段, 已有许多研究对其展开探索. 文献[1]描述了不同的通孔尺寸对于层分配的影响. 通孔最小化在文献[2]中被证明是 NP 完全问题. 文献[3~5]使用了基于协商的对线网进行拆解和重新布线的总体布线策略. 线网排序、动态规划和

收稿日期: 2021-08-09; 修回日期: 2021-11-16; 责任编辑: 覃怀银

基金项目: 国家自然科学基金(No.61877010, No.11501114); 国家重点基础研究发展计划(No.2011CB808000); 计算机体系结构国家重点实验室开放课题(No.CARCHB202014); 福建省自然科学基金(No.2019J01243)

基于协商的设计方法已被广泛运用于层分配算法设计,同时通孔数优化已成为层分配算法的重要目标<sup>[6-12]</sup>.随着集成电路规模的不断增大,传统的总体布线算法难以在布线质量和运行时间之间取得良好的平衡<sup>[13,14]</sup>.

另外一方面,针对布线算法方面的并行机制也已提出一些研究工作<sup>[15-21]</sup>.并行技术在基于整数规划的总体布线应用中表现优秀<sup>[15]</sup>.文献<sup>[16]</sup>提出了一种并发层分配框架,能够减少布线结果陷入局部最优的风险.并行技术已被应用于矩形 Steiner 最小树 (Rectilinear Steiner Minimal Tree, RSMT) 的构建,并能够有效优化线长、内存利用率和处理时间<sup>[17,18]</sup>.处理资源冲突问题是并行布线的关键<sup>[19]</sup>.线网的布线顺序、布线边界框和冲突感知策略也被用于处理资源冲突现象<sup>[10,19-21]</sup>.

然而,传统的处理资源冲突的方法关键在于最小化冲突,并不能完全避免并行布线阶段的资源冲突.为此,本文设计了一种基于区域划分的并行策略,避免并行布线阶段所产生的布线资源冲突问题.同时,利用 3D 布线区域中线网基本有多种等效的布线方案的特点,本文设计了一种基于线网等效布线方案感知的通孔优化策略.

## 2 问题模型

总体布线和层分配问题通常运用网格图模型来描述.图 1(a)展示了一个两层的布线空间,虚线将各层的布线区域划分为更小单位的布线单元 g-cell,总体布线的任务就是合理地放置导线,以连接处于不同位置的 g-cell.为了方便地研究总体布线过程中层分配阶段的布线方案,根据图 1(a)建立 3D 网格模型如图 1(b)所示.其中节点  $v$  表示一个 g-cell,边  $e$  表示连接相邻 g-cell 的导线或通孔.其中,用于连接同一层 g-cell 的水平方向上的边表示导线,而用于连接不同层 g-cell 的垂直方向上的边表示通孔.将 3D 网格模型进一步压缩成 2D 网格模型如图 1(c)所示.其中图 1(b)中的节点  $v_{1,1}$  与  $v_{1,2}$  被压缩成图 1(c)中的节点  $v_1$ ;而其中图 1(b)中的边  $e_{1,1}$  与  $e_{1,2}$  被压缩成图 1(c)中的边  $e_1$ .层分配的工作则是基于 2D 布线方案生成 3D 布线方案,根据位于  $e_1$  处的布线请求将导线分配给 3D 网格中的  $e_{1,1}$  与  $e_{1,2}$  之一,并且分配的结果应满足层分配的约束.由于制造工艺的要求,在实际层分配过程中相邻的层所放置的导线方向上应是正交的.

为了保证可布线性,层分配应避免在每一条布线边放置超过其布线容量的导线数量而导致溢出的情况.因此,本文层分配算法通过遵循以下约束来防止布线区域过度拥塞<sup>[11]</sup>:

$$\text{TWO}(S^k) = \text{TWO}(S) \quad (1)$$

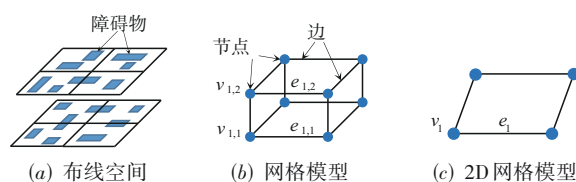


图 1 层分配的问题模型

$$\text{MWO}(S^k) = \lceil \text{MWO}(S) \times (2/k) \rceil \quad (2)$$

其中,  $S$  表示给定的 2D 总体布线结果,  $k$  表示布线空间的层数,  $S^k$  表示  $S$  的层分配结果. TWO 和 MWO 分别表示总导线溢出和最大导线溢出.在基于协商的通孔感知层分配阶段和通孔精炼阶段使用上述两个公式.式 (1) 要求线网 3D 布线方案的总导线溢出与 2D 布线方案的总导线溢出相等.另一方面,由于相邻的层上所放置的导线需要正交,  $k$  层布线空间中有  $k/2$  层放置水平方向导线,另  $k/2$  层放置垂直方向导线.因此,式 (2) 要求线网 3D 布线方案中任一层最大导线溢出等于 2D 布线方案最大导线溢出的  $2/k$ .第一个约束条件确保 3D 布线区域中的导线溢出不会超过 2D 布线区域中的导线溢出.第二个约束条件确保 2D 布线区域中边的最大导线溢出可以均匀地分布到 3D 布线区域中其相应边.

## 3 算法设计

本文所提出的面向超大规模集成电路物理设计的通孔感知的并行层分配算法将并行过程的负载均衡和布线空间的合理利用作为设计的重点,并使用协商框架来处理 3D 布线空间中的拥塞问题.

### 3.1 算法流程

算法将每个线网的 2D 布线方案视为一棵 2D 布线树,通过动态规划算法将每一棵 2D 布线树的每条边分配到 3D 布线空间中,进而获取 3D 布线方案.图 2 给出通孔感知层分配并行算法的流程.包括三个阶段:通孔主导的层分配阶段、基于协商的通孔感知层分配阶段和通孔精炼阶段.在这三个阶段中都使用了基于区域划分的并行策略(详见 3.2 节).此外,在基于协商的通孔感知层分配阶段和通孔精炼阶段都使用了等效布线方案感知的通孔优化策略(详见 3.3 节),以使布线方案在避开拥塞区域的同时减少通孔数量.

通孔主导的层分配阶段的目的是构造一个初始层分配方案.在完成布线资源的初始化后进行初始布线.本阶段中每个线网都仅考虑通孔数量,而不考虑布线区域的拥塞状况,线网的布线方案会尽量避免使用通孔.由于此阶段忽略了布线区域的拥塞状况,因此 3D 布线区域中的边容量可能会溢出.如果某个线网的布线方案经过具有溢出的边,则称该线网为违规线网.

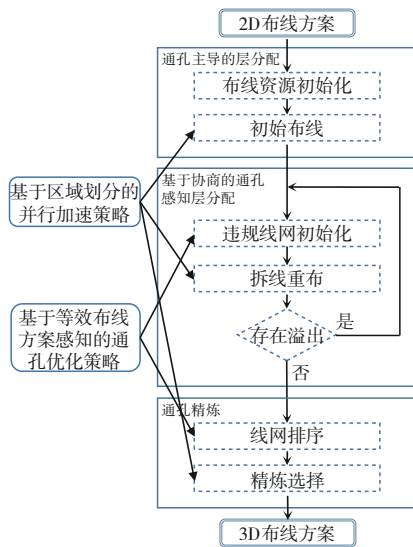


图2 算法流程

在基于协商的通孔感知层分配阶段,通过迭代的方式对违规线网进行拆解和重新布线,以减少3D布线区域中的溢出.另一方面,为了减少违规线网重新布线后的通孔数量,算法会先对所有违规线网进行排序.基于协商的通孔感知层分配阶段的相关伪代码见算法1.其中, $N_{\text{in}}$ 为违规线网集合, $N$ 为需要进行违规线网检测的集合,Checkillegal函数的功能是检测当前线网是否违规.Sort函数的功能是对线网进行排序,Partition函数的功能是区域划分,RipupRetoute函数的功能是对违规线网进行拆解和重新布线.为使线网能够避开拥塞的区域,层分配代价函数需要考虑到拥塞.首先,对于布线空间中某一条边拥塞代价设置如下:

$$\text{cost}_{\text{of}}(e) = \text{ofc}(e) \times T_e^i \quad (3)$$

其中, $\text{cost}_{\text{of}}(e)$ 为边 $e$ 的拥塞代价, $\text{ofc}(e)$ 表示边 $e$ 的溢出数, $T_e^i$ 为第 $i$ 次迭代时的历史代价.溢出数计算方式如下:

$$\text{ofc}(e) = \begin{cases} 0, & \text{dem}(e) \leq \text{sup}(e) \\ \text{dem}(e) - \text{sup}(e), & \text{otherwise} \end{cases} \quad (4)$$

算法1 基于协商的通孔感知层分配算法

输入:  $N$

输出:  $N_{\text{in}}, N$

1: LOOP

2: 初始化 $N_{\text{in}}$ ;

3: 对每个线网 $n \in N$

4:  $N_{\text{in}} \leftarrow \text{Checkillegal}(n)$ ;

5: IF  $N_{\text{in}}$  非空 THEN

6: Sort( $N_{\text{in}}$ );

7: Partition( $N_{\text{in}}$ );

8: RipupRetoute( $N_{\text{in}}$ );

9: GO LOOP

10: END LOOP

其中, $\text{dem}(e)$ 为边 $e$ 当前已放置的导线数量, $\text{sup}(e)$ 为边 $e$ 可容纳的导线数量.当 $\text{ofc}(e)$ 为0时,边 $e$ 无溢出; $\text{ofc}(e)$ 大于0时,边 $e$ 溢出.为使新一轮迭代过程中线网能够避开拥塞区域,迭代历史代价设置如下:

$$T_e^{i+1} = \begin{cases} 0, & \text{ofc}(e) = 0 \\ T_e^i + \lambda \times 2^i, & \text{ofc}(e) > 0 \end{cases} \quad (5)$$

其中, $\lambda$ 为代价系数,本文中设置为0.05.综上,设 $L_n$ 为线网 $n$ 的层分配布线方案,则其代价由以下公式计算而得:

$$\text{cost}(L_n) = v_n + \sum_{s_e \in L_n} \text{cost}_{\text{of}}(e) \quad (6)$$

其中, $v_n$ 为 $L_n$ 所使用的通孔数量, $S_e$ 为边 $e$ 上放置的导线.在每一次迭代布线过程中对各线网 $n$ ,均采用 $\text{cost}(L_n)$ 最小的方案进行布线.若拆线重布后的层分配结果中仍存在3D网格边的溢出现象,则发生溢出的3D网格边的拥塞成本都会增加,并开始新一轮的迭代,直到3D布线方案中不存在溢出的边.

在通孔精炼阶段,算法首先根据本文提出的基于线网等效布线方案感知的通孔优化策略对所有的线网进行排序,再逐一对各线网进行拆解和重新布线,进行精炼选择.若重新布线后线网的通孔数量减少,则采用重新布线后的层分配方案.否则,保留线网拆解之前的层分配方案.

### 3.2 基于区域划分的并行策略

区域划分策略通过对布线空间的划分以及线网可并行性的识别,能够避免并行布线中的资源冲突现象.此外,各并行区域的线网规模直接影响算法的并行效率.若各区域的线网规模不均衡(线网规模为所在区域内的所有线网2D布线方案长度总和),则会降低基于区域划分的并行层分配算法的效率.

#### 3.2.1 并行布线中资源冲突现象

资源冲突问题是并行算法所要处理的一个关键问题.图3展示了一个并行布线过程中的资源冲突的案例.图3(a)为一个拥有两层结构和三个线网的布线空间, $m_1$ 与 $m_3$ 为层的编号,线网1、线网2和线网3的引脚分别用蓝色、黄色和紫色的点标记.假设此时边 $e_{1,1}$ 的布线资源仅能放置一条导线.在串行布线过程中,各线网按顺序进行布线,假设线网1的布线顺序在线网2之前.如图3(b)所示在线网1占据了 $e_{1,1}$ 唯一可用的布线资源后,线网2会选择在 $m_3$ 层放置导线以绕过拥塞区域.在并行布线过程中,假设线网1和线网2由不同的两个线程同时布线.此时两个线程在布线过程中都认为边 $e_{1,1}$ 处的可用资源都为1,因此,两线网的布线结果如图3(c)和图3(d)所示.然而,如图3(e)所示,边 $e_{1,1}$ 实际放置的导线数量为2,产生了溢出.由于线网1和线网2存在溢出现象,将被定义为违规线网,在新的一轮迭代中需要

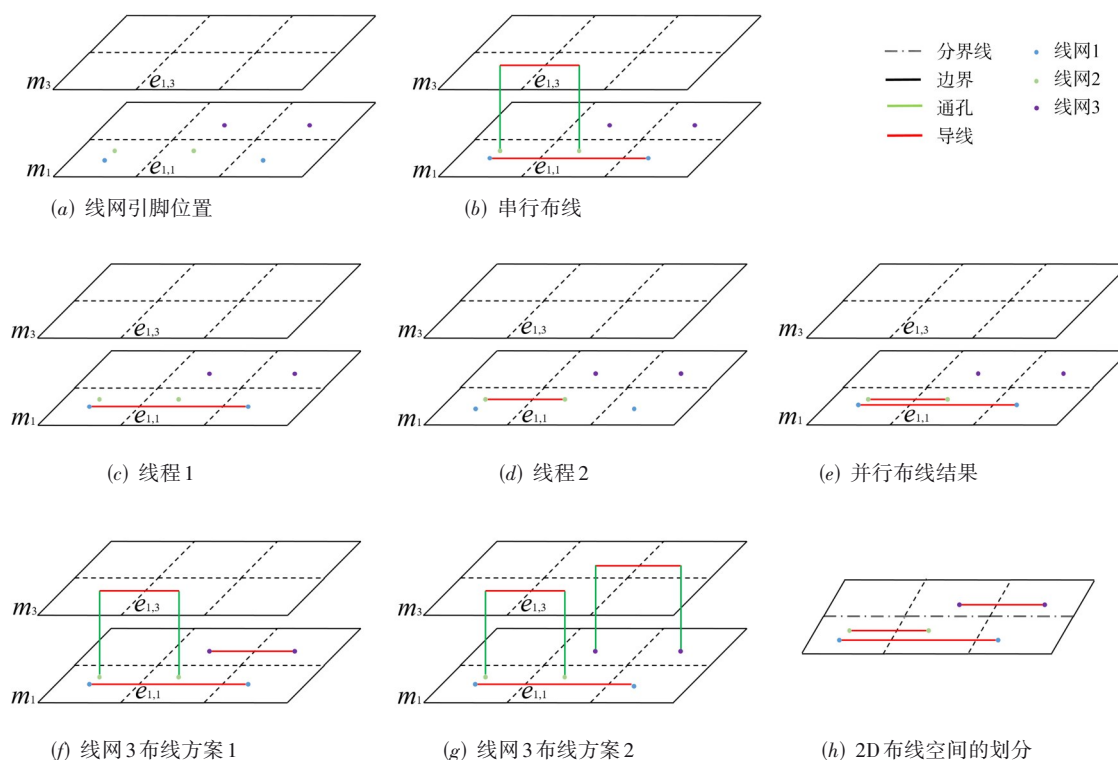


图3 资源冲突示意图

进行拆线重布. 由此可见, 资源冲突会增加需要拆线重布的线网的数量, 从而增加算法的运行时间. 图3(f)和图3(g)展示了线网3在 $m_1$ 和 $m_3$ 层两种不同的层分配方案. 无论线网3选择 $m_1$ 或 $m_3$ 层放置导线, 均不会与线网1和线网2产生资源冲突, 因为它们使用不同的3D布线资源. 如图3(h)所示为线网1、线网2和线网3在2D布线空间中的布线方案. 若将2D布线空间沿水平方向设置一条分界线, 将2D布线空间划分为上下两个部分, 上下两个区域的线网在层分配的过程中所使用的3D布线资源并不会产生冲突. 因此, 可以根据线网的2D布线方案将布线空间划分为若干个不同的区域, 让每一个区域内的线网都由一个线程进行布线, 以避免并行布线过程中发生资源冲突现象.

### 3.2.2 区域划分

区域划分如图4所示, 黑色实线的方框表示总体布线区域, 用一横一纵两条分界线将总体布线区域划分为四个并行布线区域. 将纵向分界线称为 $D_x$ , 横向分界线称为 $D_y$ . 所有的线网都可由分界线分成两类: (1) 可并行布线线网. 此类线网的2D布线图仅位于一个并行布线区域内, 在层分配过程中仅使用自身所在区域的布线资源. 因此, 位于不同并行布线区域内的线网在并行布线过程中不会出现资源冲突. (2) 不可并行布线线网. 此类线网横跨了多个并行布线区域, 在并行布线的过程中可能会出现资源冲突的现象.

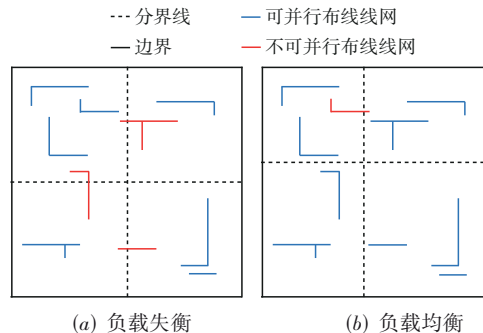


图4 区域划分示意图

在总体布线区域中, 线网并不是均匀分布的, 因此每个区域所拥有的线网规模并不由该区域的面积(可布线资源)所决定. 若 $D_x$ 与 $D_y$ 位置不合适, 将导致各区域的线网规模出现严重的失衡. 如图4(a)所示, 位于左上角并行区域的线网规模明显大于位于右上角和左下角, 在并行布线的过程中会导致右上角和左下角的并行区域相较于左上角过早地完成布线, 需要等待左上角并行布线区域完成布线的的时间过长. 因此, 若各区域在并行布线的过程中负载不均衡, 将降低并行布线的效率, 分界线需要设置在能使各并行区域负载均衡的位置. 图4(b)展示了一种相较于图4(a)更为合理的分界线设置方案.

本文提出了一种基于线网规模感知的区域划分策略, 将分界线动态地调整到合适的位置, 动态调整分界

线的原则是使四个并行区域所拥有的线网规模接近,从而让并行执行的四个子线程负载均衡.如算法 2 所示,  $R_{lu}$  为左上角的并行区域,  $R_{ru}$  为右上角的并行区域,  $R_{lb}$  为左下角的并行区域,  $R_{rb}$  为右下角的并行区域.在分界线  $D_x$  与  $D_y$  调整的初始阶段,各个区域的线网规模差异较大,称此时各区域是不均衡的.因此在迭代的初始阶段需要使用较大的幅度来对  $D_x$  与  $D_y$  进行调整,以加快各区域负载均衡的速度.随着迭代次数的增加各区域之间的线网规模会越趋于平衡,迭代过程中的调整幅度应逐步减小,以更精确地确认使各区域均衡的分界线的位置.调整幅度  $k$  的初始值可根据分界线未进行调整时各区域线网规模的差异程度或是线网的规模来设置,本文  $k$  初始值设置为  $ite$  的上界,在每次迭代过程中更新  $k$  的计算方式如下:

$$k = k - \lceil ite/\alpha \rceil \quad (7)$$

其中,  $ite$  表示当前迭代次数.  $k$  表示每次区域划分调整的幅度.  $\alpha$  是一个常量,本文中设置为  $ite$  的上界.

**算法 2 区域划分算法**

```

输入:  $D_x, D_y$ 
输出:  $D_x, D_y$ 
1: 初始化  $D_x, D_y, ite, k$ 
2: LOOP
3: IF  $((R_{lu} + R_{ru}) > (R_{lb} + R_{rb}))$  THEN
4:    $D_y += k$ ;
5: ELSE
6:    $D_y -= k$ ;
7: IF  $((R_{lu} + R_{lb}) < (R_{ru} + R_{rb}))$  THEN
8:    $D_x += k$ ;
9: ELSE
10:   $D_x -= k$ ;
11: IF  $ite$  未达其上界 THEN
12:   GO LOOP
13: END LOOP
    
```

**3.2.3 区域负载失衡现象及解决方案**

各并行布线区域负载均衡,是基于区域划分的并行布线策略能够高效运行的重要保障.区域划分算法首先在通孔主导的层分配阶段的并行层分配之前执行.但在基于协商的通孔感知层分配阶段各区域违规线网的数量是无法确定的,某些区域的违规线网规模可能过多,或过少.导致在对违规线网进行拆解和重新布线时各区域的违规线网的规模差异过大,进而导致负载失衡,降低了并行布线的效率.在经过多次拆线重布迭代之后各区域违规线网规模的差异越来越大,随着迭代次数的增加,各区域负载失衡愈加严重,并行效率随着迭代次数的增加而降低.这一现象在线网规模较大、线网数量较多的测试用例中尤为显著.因此,

仅在通孔主导的层分配阶段进行区域划分并不能保证后续每一次层分配过程中并行布线的效率.

此外,在超大规模集成电路中,一些线网的规模可能是小规模线网的十几倍甚至几十倍,布线所需的时间也是小规模线网的十几倍、几十倍.在基于协商的通孔感知层分配阶段的迭代末期,虽然需要进行拆线重布的线网的数量越来越少,但其中仍可能存在规模较大的线网,若不对其进行区域划分,可能导致多数大规模线网同时由一个子线程进行层分配,使并行的子线程间的任务负载量不均衡,从而降低了算法并行的效率.因此在基于协商的通孔感知层分配阶段,每一轮迭代都应重新进行区域划分,再进行并行布线,以解决基于协商的通孔感知层分配阶段迭代时产生的区域负载失衡问题,提高算法并行效率.

**3.3 基于线网等效布线方案感知的通孔优化策略**

影响通孔数量的关键因素是布线空间的合理使用和线网的布线顺序.在层分配阶段,线网通常有多个布线方案.图 5(a)与图 5(b)为两个 3D 布线空间中的布线案例,称图 5(a)所示的线网为线网  $a$ ,图 5(b)所示的线网为线网  $b$ .为使线网的通孔数量最小化,线网  $a$  的线网仅有如图 5(c)的一种布线方案;而线网  $b$  则拥有多种等效布线方案如图 5(d),图 5(e),图 5(f)所示,线网  $b$  拥有较多等效布线方案可以选择.若此时边  $e_1$  的布线资源只够一条导线使用,则应该将该资源让给线网  $a$  优先使用.因为线网  $b$  有不使用  $e_1$  的布线方案.若我们先将布线资源使用权分配给线网  $b$ ,而线网  $b$  选择了图 5(e)的所示的布线方案,则线网  $b$  布线完成后,  $e_1$  已经没有可用的布线资源.这会导致算法对线网  $a$  进行布线时,由于线路拥塞无法选择  $e_1$ ,因而线网  $a$  的布线方案需要使用额外的通孔以满足拥塞约束.若是先将布线资源分配给线网  $a$ ,则线网  $b$  还有图 5(d)和图 5(f)两种可行的布线方案,不

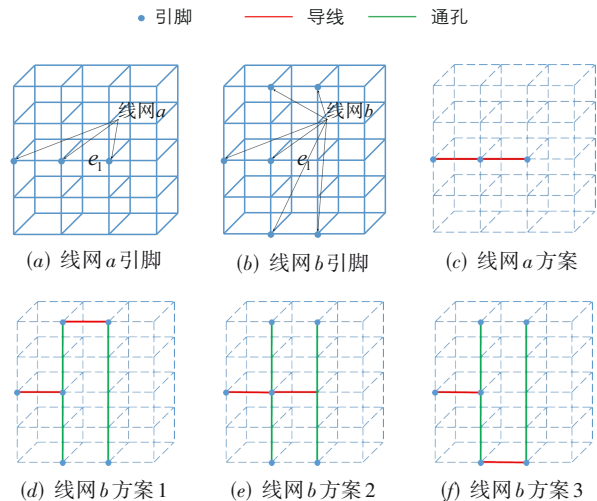


图 5 线网的 3D 布线方案实例

需要使用额外的通孔来满足拥塞约束。

在上述布线方案中线网  $a$  的通孔数量较少,各个节点之间的层数差异较小,线网  $a$  仅有一个最优布线方案;而线网  $b$  的通孔数量较多,各节点之间的层数差异较大,线网  $b$  有较多的等效布线方案.若先对拥有较多等效布线方案的线网进行布线,则等效布线方案较多的线网可能会抢占等效布线方案较少的线网的布线资源,使其产生额外的通孔。

如何评价线网的等效布线方案的数量是十分关键的.仅靠线网的通孔数量来判断线网等效布线方案数量显然不合适,还需要考虑线网在 2D 布线方案下的长度.如图 6 所示的两个线网,图 6(a) 线网的通孔数量是 4,图 6(b) 线网的通孔数量是 2.由于相邻层之间的导线方向需正交,图 6(b) 线网有如图 6(c) 的等效布线方案.图 6(a) 线网的通孔数量比图 6(b) 线网多,但图 6(a) 线网通孔最小化的等效布线方案数量比图 6(b) 线网的等效布线方案数量少。

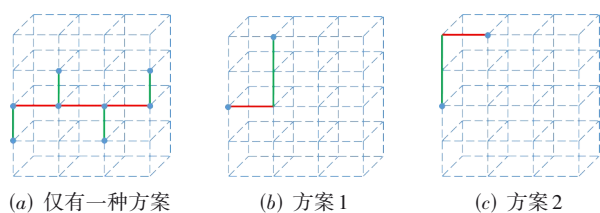


图 6 线网通孔数量、线网长度与等效布线方案的关系。

因此需要结合线网 2D 布线方案中的长度以及线网的通孔数量来评价一个线网的等效布线方案数量.为每个线网设置一个  $P_n$  值来评价线网的等效布线方案数量, $P_n$  值计算公式如下:

$$P_n = v_n / w_{2D} \quad (8)$$

其中,  $w_{2D}$  表示线网 2D 布线方案中的长度.  $P_n$  值是一个用于评价线网布线方案数量的值,  $P_n$  值越大表示线网的等效布线方案数量越多,  $P_n$  值越小表示线网的等效布线方案数量越少。

为使等效布线方案数量较少的线网能优先使用布线资源,在基于协商的通孔感知层分配阶段和通孔精炼阶段算法会根据线网  $P_n$  的值来对所有的线网进行升序排序,再进行层分配。

经过区域划分之后,布线方案会产生一部分线网跨越多个并行布线区域,这些线网无法进行并行布线.无法并行布线的线网中有一部分线网的  $P_n$  值大于并行布线区域中的多数线网,而另一部分线网的  $P_n$  值小于并行布线区域中的多数线网,并且需要在并行布线区域的线网完成并行布线之前或者之后进行布线,会破坏布线过程中的线网  $P_n$  值的有序性.因此为减小无法并行的线网在布线时对线网  $P_n$  值有序性的破坏,算法

将根据线网的  $P_n$  值将所有不可并行的线网分成两组.  $P_n$  值小于平均值的线网在并行布线之前进行布线,而  $P_n$  值大于平均值的线网则在并行布线结束之后再进行布线。

## 4 实验结果

本文所提出的层分配算法已用 C++ 语言实现,实验使用 64 核 2.80 GHz 处理器,并在 256 GB 内存的 Linux 环境中执行.实验所使用的测试电路数据源自国际设计自动化顶级会议比赛数据<sup>[22]</sup>.布局方案由 NTU-place4<sup>[23]</sup> 产生,2D 总体布线方案由 NCTU-GR2.0<sup>[20]</sup> 生成.每个测试用例的布线区域规模和需要处理的线网数量如表 1 所示.其中 #grid 反映了每个布线层中网格单元的数量, #layer 表示多层布线区域的层数, #net 表示电路中需要处理的线网数量。

表 1 测试用例特征表

benchmark	#grid	#layer	#net
superblue2	770×891	9	990899
superblue3	800×415	9	898001
superblue6	649×495	9	1006629
superblue7	499×713	9	1340418
superblue9	426×570	9	833808
superblue11	631×878	9	935731
superblue12	444×518	9	1293436
superblue14	406×473	9	619815
superblue16	465×404	9	697458
superblue19	321×518	9	511685

### 4.1 基于区域划分的并行策略加速效果的验证

基于区域划分的并行策略的有效性通过算法三个阶段的加速比和算法总运行时间的加速比四个方面来验证。

表 2 展示了各用例在通孔主导的层分配阶段的用时和加速比.  $VLA\_runtime$  表示算法在通孔主导的层分配阶段的用时,  $VR$  表示仅使用本文所提出的通孔优化策略而未使用并行策略的串行层分配算法,  $PVR$  表示面向超大规模集成电路物理设计的通孔感知的并行层分配算法,  $SUP$  表示运行时间加速比,由串行的层分配策略用时除以基于区域划分的并行策略用时求得.实验结果表明,在采用基于区域划分的并行策略的通孔主导的层分配阶段的平均加速比达到了 2.20。

表 3 展示了各用例在基于协商的通孔感知层分配阶段的用时和加速比.在采用基于区域划分的并行策略的基于协商的通孔感知层分配阶段的平均加速比达到了 2.08.基于协商的通孔感知层分配阶段需要筛选出违规线网进行拆线重布,增加了非并行布线阶段的用时.因此各测试用例在本阶段的加速比略有降低。

表 2 通孔主导的层分配阶段串行算法与基于区域划分的并行算法的运行时间对比

benchmark	VLA_runtime		SUP
	VR	PVR	
superblue2	46.407	17.438	2.66
superblue3	24.558	11.913	2.06
superblue6	22.903	11.770	1.95
superblue7	30.187	11.690	2.58
superblue9	16.459	8.159	2.02
superblue11	25.200	9.544	2.64
superblue12	26.055	11.109	2.35
superblue14	14.041	7.130	1.97
superblue16	16.737	9.407	1.78
superblue19	9.226	4.513	2.04
average			2.20

表 3 基于协商的通孔感知层分配阶段串行算法与基于区域划分的并行算法的运行时间对比

benchmark	NVLA_runtime		SUP
	VR	PVR	
superblue2	240.144	100.979	2.38
superblue3	167.022	86.836	1.92
superblue6	137.432	67.692	2.03
superblue7	177.883	74.700	2.38
superblue9	96.693	46.674	2.07
superblue11	103.781	43.040	2.41
superblue12	164.388	78.414	2.10
superblue14	85.886	40.345	2.13
superblue16	108.72	70.933	1.53
superblue19	44.396	23.626	1.88
average			2.08

表 4 展示了各用例在通孔精炼阶段的用时和加速比. 在采用基于区域划分的并行策略的通孔精炼阶段

表 4 通孔精炼阶段串行算法与基于区域划分的并行算法的运行时间对比

benchmark	VRO_runtime		SUP
	VR	PVR	
superblue2	67.037	24.493	2.74
superblue3	39.515	16.427	2.41
superblue6	37.739	15.853	2.38
superblue7	48.160	16.991	2.83
superblue9	26.944	11.225	2.40
superblue11	37.795	13.195	2.86
superblue12	40.811	17.307	2.36
superblue14	23.018	9.983	2.31
superblue16	27.575	12.868	2.14
superblue19	13.725	6.088	2.25
average			2.47

的平均加速比达到了 2.47. 在本阶段对于布线用时较多的测试用例并行策略有更好的加速效果.

表 5 展示了本文算法对各用例进行布线的用时和加速比. 算法完成各用例的布线工作的平均加速比达到了 2.16, 有着良好的整体加速效果. 此外, 本文所提出的并行算法的加速比优于文献[10]中的并行策略.

表 5 基于通孔感知的并行层分配算法整体运行时间对比

benchmark	total_runtime			SUP	
	VR	文献[10]	PVR	文献[10]	PVR
superblue2	353.588	161.582	142.910	2.19	2.47
superblue3	231.095	126.879	115.175	1.82	2.01
superblue6	198.074	110.927	95.315	1.79	2.08
superblue7	256.230	104.435	103.381	2.45	2.48
superblue9	140.096	85.728	66.058	1.63	2.12
superblue11	166.766	55.983	65.799	2.98	2.54
superblue12	231.254	107.340	106.830	2.15	2.16
superblue14	122.945	119.577	57.458	1.03	2.14
superblue16	153.032	81.969	93.208	1.87	1.64
superblue19	67.347	34.430	34.227	1.96	1.97
average				1.99	2.16

#### 4.2 基于线网等效布线方案感知的通孔优化策略有效性验证

为验证基于线网等效布线方案感知的通孔优化策略的有效性, 将仅使用本文提出的通孔优化策略的串行层分配算法(表中表示为 VR)与文献[11]、文献[12]中所提出的层分配算法进行对比, 如表 6 所示. via\_count 表示层分配算法所使用的通孔数量, DDR<sup>[11]</sup>和 DLA<sup>[12]</sup>分别为两种从不同角度考虑通孔优化的层分配算法. ratio 表示通孔数量优化百分比. 与 DDR 和 DLA 进行对比, 通孔数量平均优化百分比分别达到了 15.98% 和 12.76%, 并且在所有测试用例中最高优化比率分别达到 20.00% 和 16.06%, 最低值分别达到 12.36%

表 6 基于线网等效布线方案感知的通孔优化策略验证

benchmark	via_count			ratio	
	DDR <sup>[11]</sup>	DLA <sup>[12]</sup>	VR	DDR <sup>[11]</sup>	DLA <sup>[12]</sup>
superblue2	7129655	6794969	5703736	20.00%	16.06%
superblue3	6382992	6094468	5276904	17.33%	13.41%
superblue6	6202333	5993170	5316640	14.28%	11.29%
superblue7	9335272	9008199	7785024	16.61%	13.58%
superblue9	5120143	4928392	4248455	17.02%	13.80%
superblue11	5592279	5463358	4692009	16.10%	14.12%
superblue12	8500960	8180039	7321668	13.87%	10.49%
superblue14	4038943	3917782	3539881	12.36%	9.65%
superblue16	4122687	3899307	3371064	18.23%	13.55%
superblue19	3038941	2960095	2614623	13.96%	11.67%
average				15.98%	12.76%

和 9.65%。实验结果表明本文所提出的通孔优化策略有着显著的通孔优化效果。

### 4.3 面向超大规模集成电路物理设计的通孔感知的并行层分配算法有效性的验证

如表 7 所示,实验结果表明虽然区域划分破坏了 3.3 节中所描述的线网  $P_n$  值的有序性,通孔优化策略的效果会被一定程度上削弱,但在通孔数优化上仍然分别达到了 11.60% 和 8.19% 的平均优化百分比,有着显著的通孔数优化效果。

表 7 本文算法的有效性验证

benchmark	via_count			ratio	
	DDR <sup>[11]</sup>	DLA <sup>[12]</sup>	PVR	DDR <sup>[11]</sup>	DLA <sup>[12]</sup>
superblue2	7129655	6794969	5885709	17.45%	13.38%
superblue3	6382992	6094468	5494079	13.93%	9.85%
superblue6	6202333	5993170	5532920	10.79%	7.68%
superblue7	9335272	9008199	7884639	15.54%	12.47%
superblue9	5120143	4928392	4268805	16.63%	13.38%
superblue11	5592279	5463358	4909798	12.20%	10.13%
superblue12	8500960	8180039	7807668	8.16%	4.55%
superblue14	4038943	3917782	3688443	8.68%	5.85%
superblue16	4122687	3899307	4105378	0.42%	-5.28%
superblue19	3038941	2960095	2668770	12.18%	9.84%
average				<b>11.60%</b>	<b>8.19%</b>

## 5 结语

在超大规模集成电路设计中,层分配是布线阶段的关键环节。层分配过程中所使用的通孔数量和运行时间是评价层分配策略与算法效率的重要指标。通孔数量优化能够减少制造成本,而减少运行时间则能够提高算法的效率。为此,本文提出面向超大规模集成电路物理设计的通孔感知的并行层分配算法,所采用的基于区域划分的并行策略和基于线网等效布线方案感知的通孔优化策略分别对运行时间和通孔数量带来良好的优化效果。

### 参考文献

- [1] SHI D, TASHJIAN E, DAVOODI A. Dynamic planning of local congestion from varying-size vias for global routing layer assignment[J]. IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems, 2017, 36(8): 1301-1312.
- [2] NACLERIO N J, MASUDA S, NAKAJIMA K. The via minimization problem is NP-complete[J]. IEEE Transactions on Computers, 1989, 38(11): 1604-1608.
- [3] MCMURCHIE L, EBELING C. Pathfinder: A negotiation-based performance-driven router for FPGAs[C]//Proceedings of the 1995 ACM Third International Symposium on Field-Programmable Gate Arrays. New York: ACM Press, 1995: 111-117.
- [4] ZHANG X H, ZHUANG Z, LIU G G, HUANG X, LIU W H, GUO W Z, WANG T C. MiniDelay: Multi-strategy timing-aware layer assignment for advanced technology nodes [C]//2020 Design, Automation & Test in Europe Conference & Exhibition. Los Alamitos: IEEE Computer Society Press, 2020: 586-591.
- [5] LIU G G, ZHANG X H, GUO W Z, HUANG X, LIU W H, CHAO K Y, WANG T C. Timing-aware layer assignment for advanced process technologies considering via pillars[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Los Alamitos: IEEE Computer Society Press, 2021: 1-14.
- [6] DAI K R, LIU W H, LI Y L. NCTU-GR: Efficient simulated evolution-based rerouting and congestion-relaxed layer assignment on 3-D global routing[J]. IEEE Transactions on Very Large Scale Integration Systems, 2012, 20(3): 459-472.
- [7] LEE T H, WANG T C. Congestion-constrained layer assignment for via minimization in global routing[J]. IEEE Transactions on Computer Aided Design of Integrated Circuits & Systems, 2008, 27(9): 1643-1656.
- [8] LI Z, ALPERT C J, HU S, MUHMUD T, QUAY S T, VILLARRUBIA P G. Fast interconnect synthesis with layer assignment[C]//Proceedings of International Symposium on Physical Design. New York: ACM Press, 2008: 71-77.
- [9] HU S, LI Z, ALPERT C J. A faster approximation scheme for timing driven minimum cost layer assignment[C]//Proceedings of International Symposium on Physical Design. New York: ACM Press, 2009: 167-174.
- [10] LIU D, YU B, CHOWDHURY S, PAN D Z. TILA-S: Timing-driven incremental layer assignment avoiding slew violations[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2017, 37(1): 231-244.
- [11] HAN S Y, LIU W H, EWETZ R, KOH C K, CHAO K Y, WANG T C. Delay-driven layer assignment for advanced technology nodes[C]//Asia and South Pacific Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2017: 456-462.
- [12] AO J C, DONG S Q, CHEN S, GOTO S. Delay-driven layer assignment in global routing under multi-tier interconnect structure[C]//Proceedings of the 2013 AcM International Symposium on International Symposium on Phys-

ical Design. New York: ACM Press, 2013: 101-107.

- [13] WU T H, DAVOODI A, LINDEROTH J T. GRIP: scalable 3D global routing using integer programming[C]//2009 46th ACM/IEEE Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2009: 320-325.
- [14] LIU J W, PUI C W, WANG F Z, YOUNG E. CUGR: detailed-routability-driven 3D global routing with probabilistic resource model[C]//2020 57th ACM/IEEE Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2020: 1-6.
- [15] WU T H, DAVOODI A, LINDEROTH J T. A parallel integer programming approach to global routing[C]//IEEE Proceedings of Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2010: 194-199.
- [16] JIANG Y J, FANG S Y. COALA: Concurrently assigning wire segments to layers for 2D global routing[C]//2020 IEEE/ACM International Conference On Computer Aided Design. Los Alamitos: IEEE Computer Society Press, 2020: 1-8.
- [17] LATHA N R, PRASAD G R. Performance and memory-efficient parallel computing framework for RSMT construction[C]//DAS H, PATNAIK P K, RAUTARAY S S, LI K C. Progress in Computing, Analytics and Networking. Singapore: Springer, 2020: 369-381.
- [18] SHYAMALA G, PRASAD G R. An efficient parallel computing framework for over the obstacle VLSI routing [C]//DAS H, PATNAIK P K, RAUTARAY S S, LI K C. Progress in Computing, Analytics and Networking. Singapore: Springer, 2020: 383-395.
- [19] HE J Y, BURTSCHER M, MANOHAR R, PINGALI K. SPRoute: A scalable parallel negotiation-based global router[C]//2019 IEEE/ACM International Conference on Computer-Aided Design. Los Alamitos: IEEE Computer Society Press, 2019: 1-8.
- [20] LIU W H, KAO W C, LI Y L, CHAO K Y. NCTU-GR 2.0: Multithreaded collision-aware global routing with bounded-length maze routing[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2013, 32(5): 709-722.
- [21] HAN Y D, ANCAJAS D M, CHAKRABORTY K, ROY S. Exploring high throughput computing paradigm for global routing[C]//2011 IEEE/ACM International Conference on Computer-Aided Design. Los Alamitos: IEEE Computer Society Press, 2011: 298-305.
- [22] VISWANATHAN N, ALPERT C, SZE C, LI Z, WEI Y G. The DAC 2012 routability-driven placement contest

and benchmark suite[C]//Proceedings of the 2012 Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2012: 774-782.

- [23] HSU M K, CHEN Y F, HUANG C C, CHEN T C, CHANG Y W. Routability-driven placement for hierarchical mixed-size circuit designs[C]//2013 50th ACM/EDAC/IEEE Design Automation Conference. Los Alamitos: IEEE Computer Society Press, 2013: 1-6.

#### 作者简介



刘耿耿 男, 1988年生, 福建南安人, 博士, 副教授, 博士生导师, 主要研究方向为EDA算法研究、计算智能及其应用。

E-mail: liugenggen@fzu.edu.cn



李泽鹏 男, 1998年生, 福建安溪人, 硕士研究生, 主要研究方向为EDA算法研究。

E-mail: li\_zepeng@yeah.net



郭文忠 男, 1979年生, 福建惠安人, 博士, 教授, 主要研究方向为计算智能及其应用。

E-mail: guowenzhong@fzu.edu.cn



陈国龙 男, 1965年生, 福建莆田人, 博士, 教授, 主要研究方向为人工智能、网络信息安全。

E-mail: cgl@fzu.edu.cn



徐宁(通讯作者) 男, 1968年生, 湖北武汉人, 博士, 教授, 主要研究方向为FPGA物理设计、电子设计自动化、大数据分析与人机智能、图像处理。

E-mail: xuning@whut.edu.cn