

门阵列和标准单元布图中 一种有效的走线道分配算法*

洪先龙

黄劲

(清华大学计算机科学与技术系, 北京 100084)

(中国电子科技大学, 成都 610054)

摘要 本文提出了一种用于门阵列和标准单元版图自动设计中的走线道分配算法—FARM, 它连接总体布线和通道布线. 算法目标是使通道的最大密度最小, 同时考虑减小线网长度和通孔数. FARM 由两部分组成: 多行走线道分配和单行走线道分配. 它已用 C 语言在 DEC 工作站和 Sun 工作站上实现, 并已用于我们开发的双层 CMOS 门阵列布图系统 MALS3 中, 实验例子表明, 它与 TimberWolf 5.6 的结果相当或更好.

关键词 走线道分配, 总体布线, 集成电路布图, 门阵列, 标准单元.

在门阵列和标准单元布图设计中, 为了解决跨行线网的连接, 我们在设计库单元时, 专门在单元上留有若干条走线道(Feed-through). 跨行线网的连线从这一行通过单元上的走线道与另一行连接. 总体布线算法只是确定了每一个线网在总体布线图(GRG)上的路径, 它可能是一棵树. GRG 上的水平边表示通道, 垂直边表示一组单元上的走线道^[1-3]. 为了连接通道布线, 我们还必须把总体布线确定的垂直方向的路径具体分配给某个单元上的走线道. 走线道分配算法就是为了完成这个任务而设计的. 通常, 在分配走线道时要考虑它的目标.

对门阵列布图模式, 我们在分配走线道时考虑在布通前提下尽量使得通道最大密度最小. 对标准单元或门海模式, 则考虑使得总的通道密度和最小. 但是, 使得通道最大密度最小不一定能导致连线总长最小. 图 1 给出了走线道分配示意图. (a) 中 U 为上单元行, B 为下单单元行, M 为中向单元行, C_1 和 C_2 是通道. U 和 B 中相同数字表示一个线网, 它们需要跨过中间单元行 M . 为此我们必须把 M 中虚线表示的走线道分配给相应线网. (b) 给出了总体布线示意图. U 中 5 个引线端被映射到节点 V_1 , B 中引线端被映射到 V_2 . 因此总体布线后这些线网表示成线段 $(V_1; V_2)$. (c) 是一种走线道分配方案, M 中虚线旁边数字表示此走线道被分配给相应线网, 最大通道密度为 3, 通道密度之和为 5. (d) 是另一种方案, 最大通道密度和密度之和都减少 1. 它们的连线长度相同. 显然, 方案(d)要优于方案(c).

以前发表的一些总体布线算法同时生成总体布线树和分配走线道^[4-7]. 它们的目标是

* 本文 1993-11-01 收到, 1994-02-21 定稿

作者洪先龙, 1940 年生, 教授, 主要研究领域为 VLSI 自动布图和高层次综合. 黄劲, 1968 年生, 博士生, 主要研究领域为 VLSI 布图.

本文通讯联系人: 洪先龙, 北京 100084, 清华大学计算机科学与技术系

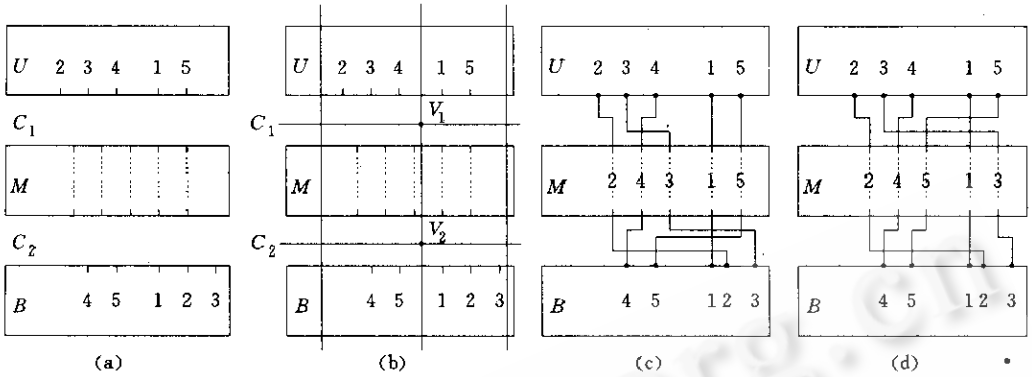


图1 走线道分配示意图

使连线总长最小和通孔数最小, 本文提出的走线道分配算法以使最大通道密度最小或通道密度和最小作为目标, 在此前提下再试图减小连线总长和通孔数.

FARM 包括两部分: 多行走线道分配和单行走线道分配. 在多行分配中, 我们在线网跨越的所有单元行中搜索具有相同 X 坐标的走线道, 把这些走线道分配给这个线网的垂直路径, 以尽可能多地消除水平段路径. 在单行分配中, 我们先对所有线网进行分类, 然后确定各个线网重直段的顺序和位置, 应用某些启发式技术交换它们的顺序和移动它们的位置以减小最大通道密度.

FARM 可用于门阵列和标准单元布图模式, 只要它们的总体布线图是规则矩形网格的图.

1 总体布线树的映射

通常, 总体布线是在一个各向等距网格的规则图 $GRG = \langle V, E \rangle$ 上完成的, 其中 $V = \{v_1, v_2, \dots, v_n\}$ 为顶点集, $E = \{e_1, e_2, \dots, e_m\}$ 为相邻顶点间的边集. 如用 e_h 和 e_v 分别表示水平和垂直方向的边, 则

$$\begin{aligned} |e_{hi}| &= |e_{hj}|, i \neq j \\ |e_{vi}| &= |e_{vj}|, i \neq j \end{aligned}$$

GRG 上的水平边表示通道, 左右两侧垂直边表示相应的左右通道, 其它垂直边表示跨越单元行走线道. 由于是等距网格, 因此在总体布线时, 可能不止一个单元上的引线端映射到同一个 v_i 上, 多个单元上的走线道映射到一个 e_{vj} 上, 这不利于走线道分配. 我们重新建立一个基于单元信息的总体布线图, 称之为布图规划图 FPG (Floorplan Graph). 它是一个不规则的图, 不一定满足 $|e_{hi}| = |e_{hj}|, i \neq j$ 的关系. 每个垂直边 e_{vi} 对应一个单元, 因此, 仅是同一个单元内的走线道映射到一个垂直边上. 它给走线道分配带来了方便. 图 2(a) 是原始总体布线图 GRG, 其上有一个对应一个线网的总体布线树. 图 2(b) 是与之对应的 FPG 及相应的总体布线树. 我们把 GRG 上的总体布线树映射到 FPG 上.

在树映射算法中, 我们首先映射顶点, 即将 GRG 上的 v_i 映射到 FPG 上的 v'_i . 然后映射树上的每一边. 在映射边时考虑到 FPG 中垂直边的走线道容量限制和最短路径. 限于论文篇幅, 映射算法的细节就不在此多述了.

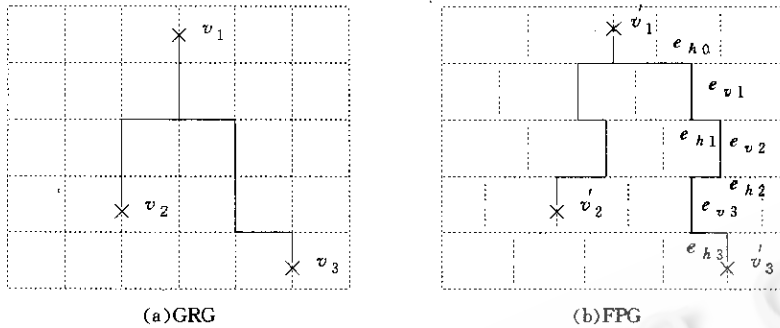


图2 GRG及其对应的FPG

2 多行走线道分配

从图 2(b) 我们可以看到, 一个总体布线树通常跨越若干单元行, 如 (v'_1, v'_3) 跨越三个单元行, 即包含了三个垂直边 e_{v1}, e_{v2} 和 e_{v3} . 而使最大通道密度最小也涉及多行. 因此, FARM 先进行多行走线道分配的操作. 多行走线道分配算法是基于尽可能多地消除两相邻垂直边之间的水平段. 如以图 2(b) 中 v'_1 到 v'_3 的路径为例, 若能将 e_{v1} 和 e_{v2} 分配在各自相应单元中具有相同 x 坐标的走线道上, 则它们之间的 e_{h1} 可以被消去, 消去 e_{h1} 将使相应通道局部密度减 1.

下面我们先对各种水平段进行分类, 然后讨论如何消除它们的技术.

2.1 水平段分类

定义 1. 总体布线树上两相邻垂直边之间的一段路径叫水平段, 记为 HS . 即水平段两端分别与两个垂直边连接. 如图 2(b) 中 e_{h0}, e_{h1}, e_{h2} 和 e_{h3} 等.

定义 2. 可以通过分配两相邻垂直边于相同 x 坐标的走线道而被消除的水平段叫可消去的水平段, 记为 RHS .

根据连接 RHS 两端的垂直边的不同, RHS 分为三类: 自由 RHS 、固定 RHS 和半固定 RHS .

定义 3. 如果连接 RHS 两端的垂直边都是与一个固定的引线端相连, 则称这个 RHS 为固定的 RHS .

定义 4. 如果连接 RHS 两端的垂直边, 只有其中一边是连接引线端, 则称为半固定 RHS .

定义 5. 如果连接 RHS 两端的垂直边都不是和引线端相连, 则称为自由 RHS .

3 种 RHS 如图 3 所示.

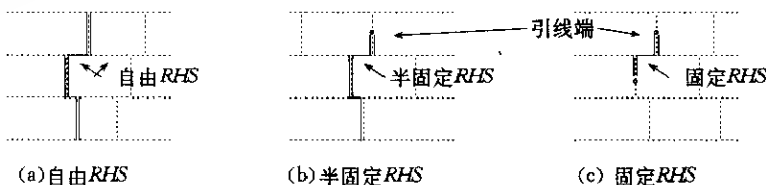


图3 3种RHS

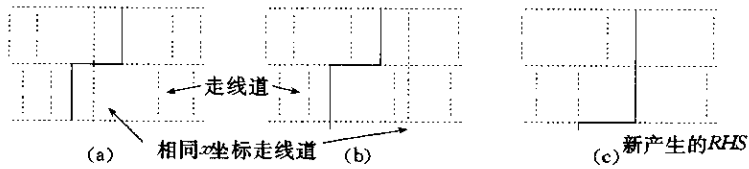


图4 自由RHS的消去

2.2 自由 RHS 的消去

由于连接自由 RHS 的两个垂直边可以被分配在它们相应单元甚至相邻单元内的任何未被占用的走线道上,因此,首先在它们所属的单元内搜索有相同 x 坐标的走线道,如果存在,则把这两个走线道分别分配给这两个垂直边,如图 4(a)所示,此自由 RHS 将消失. 如果不存在有相同 x 坐标的走线道,则扩大搜索范围至它们左右相邻的单元中. 若存在,可考虑分配这两个走线道给它们(图 4(b)),但此时会产生一个额外的 RHS,如图 4(c)所示. 如果此新产生的 RHS 不影响所在通道的最大密度,则接受这个分配;否则,不进行这个分配,保留原来的 RHS.

2.3 半固定 RHS 的消去

半固定 RHS 一端连着一个引线端,引线端在单元中有固定位置,不存在有走线道分配问题,但可利用它在单元上下有电等价端的性质向下或向上引出. 半固定 RHS 另一端是一个自由垂直边,可对它进行走线道分配. 首先我们在自由边所属单元中寻找与另一侧引线端有相同 x 坐标的走线道,如图 5(a)所示,存在这样的走线道. 我们分配这个走线道给自由边,则可消去此 RHS. 否则,我们在上下两行相应的单元内寻找具有相同 x 坐标的走线道,如图 5(b)所示,若存在这样的走线道,分配给它们走线道并让固定端反方向连接走线道. 此时,可消去原来的 RHS,但产生了一个新的 RHS. 若此新 RHS 不影响所在通道的最大密度,则接受这个分配;否则,保留原来的 RHS. 类似于自由 RHS 的消去,我们也可在左右相邻单元中寻找有相同 x 坐标的走线道,并分配给上下两个垂直边. 同样,对固定端需反向引出并增加一个新 RHS. 是否接受这个分配,则取决于新 RHS 是否会增加最大通道密度(图 5(c)).

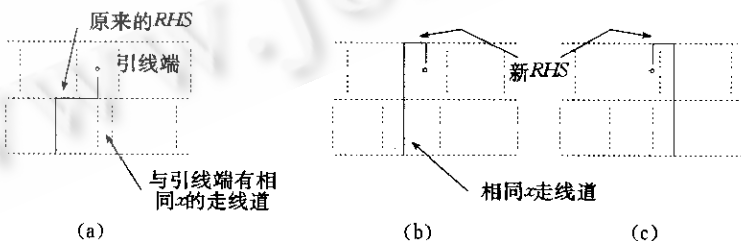


图5 半固定RHS的消去

2.4 固定 RHS 的消去

由于固定 RHS 的两端均连接着引线端,它们本身的位置已经固定,只能利用一端或两端反向连接技术消去原始的 RHS,但会增加新的 RHS. 如新产生的 RHS 不影响通道密度,则可采用这种消除,否则仍保留原来的 RHS. 图 6 表示了一端和两端反向连接消去原始

RHS 的实例.

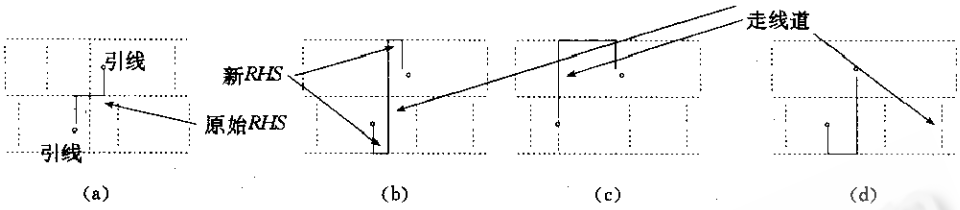


图6 固定RHS的消去

2.5 多行RHS的消去

我们在前面已介绍了跨越两个单元行的总体布线树中 RHS 的消除. 实际上, 许多树可能跨越更多单元行. 如图 7(a)所示的布线树跨越了 5 个单元行, 包含了 4 个 RHS, 圆点表示的线为走线道. 我们搜索相邻两单元行中具有相同 x 坐标的走线道, 并把它们分配给总体布线树中的垂直边, 尽可能多地消除 RHS. 对于多行的 RHS 消除, 可能存在多种方案, 如图 7(b)和图 7(c). 我们枚举所有可能方案, 寻找一种使最大通道密度最小的方案. 如(b)中消去 RHS_2 和 RHS_4 , (c)中消去 RHS_1 和 RHS_3 , 比较哪一种方案更有利于减小最大通道密度.

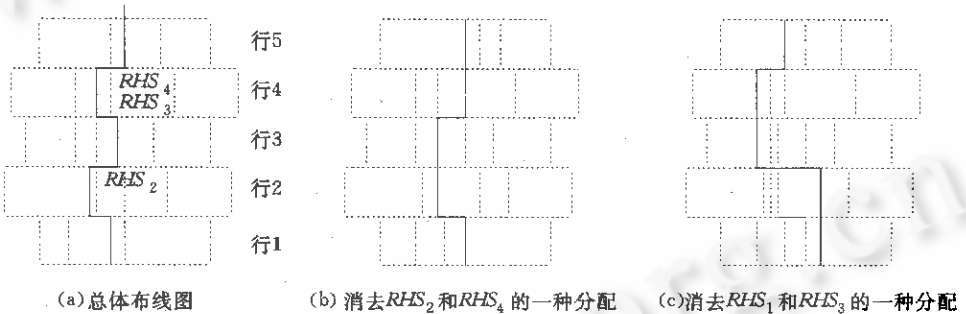


图7 多行RHS的消去

2.6 多行走线道分配算法

我们根据在第 1 节介绍的总体布线树的映射, 把 GRG 上的布线结果映射到 FPG 上. 然后估计各区域总体布线的局部通道密度和总体最大通道密度.

对门阵列布图模式, 我们每次都寻找最大通道密度区, 在这个区域寻找跨越该区的总体布线树, 执行多行 RHS 消除和单个 RHS 消去算法. 然后更新通道密度值, 直到最大密度不能下降, 算法停止. 下面给出这个算法的描述:

- (1) 计算基于 FPG 上总体布线结果的各通道段的初始通道密度;
- (2) 寻找跨越最大密度区的线网, 设为 A ;
 若 A 跨越两个以上单元行, 则执行多行 RHS 消除算法;
 若 A 仅跨越两个单元行, 分三种情况:
 若 A 为自由 RHS, 执行自由 RHS 消去;
 若 A 为固定 RHS, 执行固定 RHS 消去;

若 A 为半固定 RHS , 执行半固定 RHS 消去;

(3)更新芯片上通道密度值;

(4)若最大密度不再下降, 停止算法; 否则重复(2), (3)步骤.

对标准单元布图模式, 算法目标为使总的通道密度和最小. 把上述算法中寻找芯片最大通道密度区更换成每个通道的最大密度区. 在 RHS 消除算法中的决定要否消去的判据改为是否有利于减小本通道的最大密度. 整个算法的终结条件为总的通道密度和不再下降.

3 单行走线道分配

多行走线道分配算法只处理那些跨越 2 行和 2 行以上单元行的线网总体布线树. 还存在不少只跨越一个单元行的线网的总体布线树. 为叙述方便, 下面所指的线网都是指线网的总体布线树. 我们在文献[8]的基础上提出一种单行走线道分配算法给这些线网分配走线道. 算法目标仍然是尽可能地使通道最大密度最小, 同时兼顾使连线长度和通孔数最小.

3.1 跨越单个单元行线网分类

我们根据线网的形状把跨越单元行的线网分成图 8 所示的 64 类. 其中 $(x0)$ 和 $(0x)$ 类型线网, 其一端和单元上的固定引线端相连, 它们不参加走线道分配操作. $(2x)$ 和 $(x2)$ 类线网的“2”端和另一相邻单行的 $(x2)$ 或 $(2x)$ 类线网相接, 实际上构成了跨多个单元行的线网. 这种线网我们在多行走线道分配中已处理过, 因此它们也不在本节处理. 剩余类型的线网, 我们将用本节叙述的单行走线道分配算法处理.

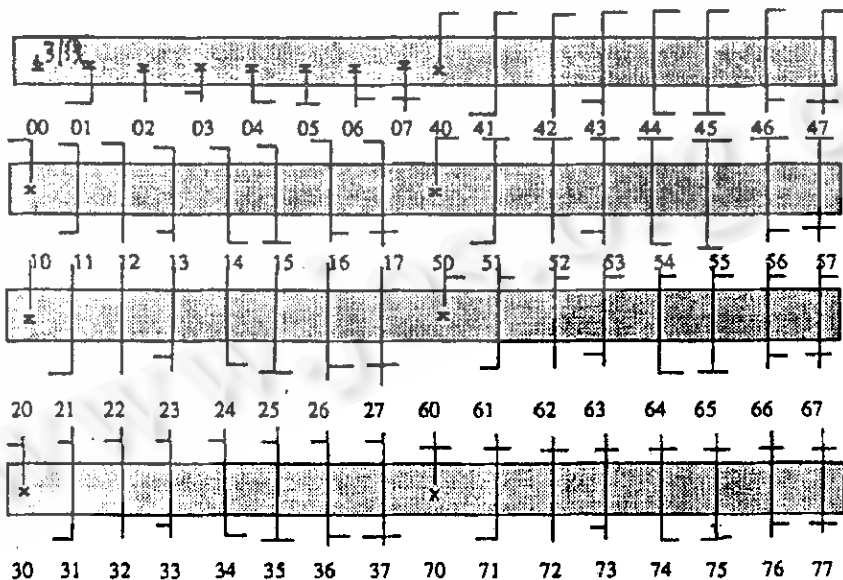


图 8 线网分类

3.2 线网的顺序约束和分区

我们按照线网形状对线网建立顺序约束. 例如(11)类型线网应位于(44)类型线网的左方, 这样, (11)和(44)线网两端的水平段可占用通道中的同一轨道. 而(14)、(41)、(16)和

(61)类型的线网则可以任意顺序放置而不影响所占用的轨道数. 我们根据尽可能少地占用轨道数的目标定义顺序约束:

$$(11) \rightarrow \begin{matrix} (13)(15)(17) \\ (31)(51)(71) \end{matrix} \Rightarrow \begin{matrix} (45)(46)(47) \\ (54)(64)(74) \end{matrix} \rightarrow (44)$$

其中 (xx) 表示节点,“ \rightarrow ”和“ \Rightarrow ”表示约束关系,“ \rightarrow ”表示软约束,“ \Rightarrow ”表示强约束. $(11) \rightarrow (13)$ 表示 (11) 类型线网应放在 (13) 左边,否则会增加通孔数. $(11) \Rightarrow (44)$ 表示 (11) 线网应放在 (44) 左边,否则会增加轨道数、线网长度及通孔数. 未在约束关系里出现的线网类型不受顺序的约束,它的走线道分配不影响通道密度,也不会影响线长及通孔数的变化.

根据顺序约束我们把每个单元分成三个区:左区、中区和右区. 每个区可包含若干走线道,但也可为空(即不含有走线道,甚至此区空缺). 通常 (11) 、 (13) 、 (15) 、 (17) 、 (31) 、 (51) 和 (71) 类型线网分配在左区, (44) 、 (45) 、 (46) 、 (47) 、 (54) 、 (64) 和 (74) 线网分配在右区, (14) 、 (34) 、 (43) 、 (55) 、 (77) 、 (57) 、 (75) 和 (41) 线网分配在中区. 图 9 给出了三个区及分配的线网类型.

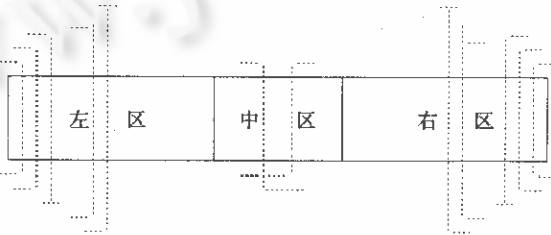


图9 分区

3.3 (11)和(44)类型线网的平移

对 (11) 线网,如果它左边相邻单元有多余的走线道未被占用,此时可将此 (11) 型线网移至左边单元分配走线道. 这可减小连线长度. 同理,也可考虑将 (44) 线网移至右边相邻单元中分配走线道. 如果相邻两单元中左单元有一个 (44) 线图,右单元有一个 (11) 线图,则它们恰好可以交换.

3.4 交换交叉点

如图 10 所示,中区的 (14) 和 (41) 线网可以通过改变它们的顺序而交换交叉点. 用这个方法可以控制局部通道密度. 有交叉点的哪一个通道多占用一个轨道.

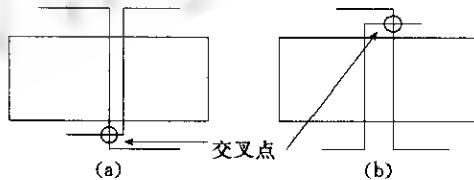


图10 交换交叉点

3.5 确定左区和右区中线网的顺序

通过分析和顺序约束操作,我们已把 (11) 、 (13) 、 (15) 、 (17) 、 (31) 、 (51) 和 (71) 分配在左区, (44) 、 (45) 、 (46) 、 (47) 、 (54) 、 (64) 和 (74) 分配在右区,但还没有确定它们的顺序. 本节我们将讨论如何确定它们的顺序,算法的目标是尽可能地减小连线长度和通孔数.

设 C 为一个圆, C_u 为其上半圆, C_d 为其下半圆, Dim 是两半圆的水平直径. 图 11(a) 是原始问题, 数字表示线网号, 相同线网号属于同一线网, 它们必须相连. 点线表示走线道. 我们将走线道映射到图 11(b) 的水平直径上, 上通道边界上数字(代表引线端或连接端)映射到上半圆周 C_u , 并按顺时针方向排列. 下通道边界上数字映射到 C_d , 并按逆时针方向排列. 如图 11(b) 中所示, 上下两半圆中相同线网号相连并经过 Dim 上表示走线道的点, 寻找一个最小相互相交的连接匹配解. 连线与 Dim 交点顺序表示了图 11(a) 中走线道顺序的分配. 图 11(c) 是图 11(a) 的最佳解. 不相交的线网利用横竖非严格分层的通道布线算法不会产生穿孔. 图 11(b) 中上半圆和下半圆的交点表示在上或下通道内必须有一个穿孔.

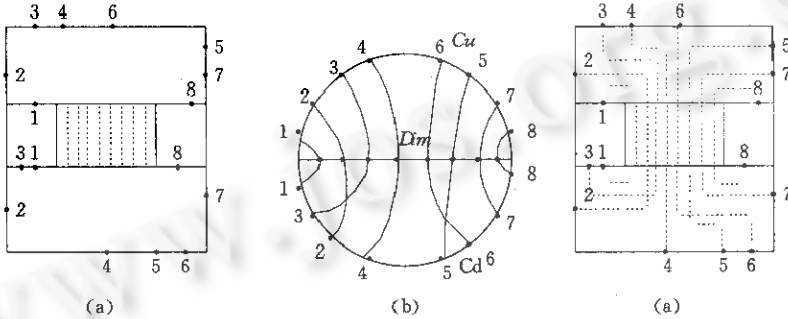


图11 左区和右区中线网顺序确定

至此, 我们已确定了所有线网垂直边的顺序, 如果单元内没有多余的走线道, 则垂直边顺序和走线道顺序一一对应, 它们的位置也就确定了. 如果走线道数大于线网数, 则可以从左右两端向中心逐个排列并确定它们的位置.

3.6 单行走线道分配算法

- (1) 从下到上、从左到右逐个扫描每个单元
- (2) 对每个单元上所有跨越单行线网执行
 - (2.1) 线网分类;
 - (2.2) 分区和建立顺序约束;
 - (2.3) 对(1 1)和(4 4)型线网作平移处理;
 - (2.4) 交换交叉点;
 - (2.5) 确定左、右区中线网顺序;
 - (2.6) 分配具体走线道;

4 实验结果

我们用 C 语言在 DEC 和 Sun 工作站上实现了 FARM. 已发表的论文中还没有见到有单独走线道分配算法的实例和结果. TimberWolf 是美国柏克利大学研制的用于标准单元和门阵列的布局 and 总体布线(包括了走线道分配算法)程序. 它采用了模拟退火算法, 计算时间很长但结果很好. 国际上绝大多数已发表的关于布局和布线的算法都与它进行比较, 以说明算法的优劣. 我们采用了 TimberWolf 5.6 的布局结果, 并和文献[1]的总体布线连接以测试 FARM 的性能. 我们用它们测试了几个国际 CAD 界公认的标准测试例子. 由于没有单独走线道分配算法的结果可供比较, 我们只得与 TimberWolf 5.6 的总体布线(包括了走线道分

配)结果进行比较^[4].表1列出了两者的结果.从表1我们可以看到,文献[1]和FARM的结果比TimberWolf 5.6的好或者相当.

表1

测试实例	单元数	线网数	TimberWolf 最大通道密度	[1]+FARM 最大通道密度
Primary1	750	915	7	7
Primary2	2907	3050	13	12
C ₂	590	963	12	12
C ₇	2150	2465	14	13

5 结论

本文提出了一种用于门阵列和标准单元布图的一种走线道分配算法.它的目标是使最大通道密度最小或者通道密度和最小,同时考虑尽可能减小线网连线长度和通孔数.算法已用于我们开发的双层CMOS门阵列布图系统中.实验结构表明,它比TimbtrWolf 5.6的结果好或者相当.

致谢 本文是根据文献[9]改写而成的,其主要工作是作者在美国加州柏克利大学访问时完成的,得到了葛守仁教授的关心和指导.

参考文献

- 1 Carden R C, Cheng C K. A global router using an efficient approximate multi-commodity multi-terminal flow algorithm. In: ACM/IEEE eds. 28th Design Automation Conference Proceedings, San Francisco, 1991:316-319.
- 2 Chang M S C, Wong C K. Global routing based on Steiner min-max trees. IEEE Transaction on CAD, 1990,9(12):1318-1320.
- 3 Chao T H, Hsu Y C. Rectilinear Steiner tree construction by local and global refinement. In: ACM/IEEE eds. International Conference on CAD-90, Santa Clara, 1990.
- 4 Lee K W, Sechen C. A global router for sea-of-gates circuits. In: EDA Association/IEEE eds. The European Design Automation Conference, Edinburgh, 1991.
- 5 Cong J. Pin assignment with global routing. In: ACM/IEEE eds. International Conference on CAD-89, Santa Clara, 1989.
- 6 Meixner G, Lauther U. A new global router based on a flow and linear assignment. In: ACM/IEEE eds. International Conference on CAD-90, Santa Clara, 1990.
- 7 Parng T M, Tsay R S. A new approach to sea-of-gates global routing. In: ACM/IEEE eds. International Conference on CAD-89, Santa Clara, 1989.
- 8 应昌胜,洪先龙.双金属层门阵列跨单元行布线问题与算法.半导体学报,1992,13(10):629-632.
- 9 Hong X L, Huang J, Kuh E S. FARM-an efficient feed-through pin assignment algorithm. In: ACM/IEEE eds. 29th Design Automation Conference, Los Angeles, 1992. 33.3-33.6.

AN EFFICIENT FEED—THROUGH ASSIGNMENT ALGORITHM FOR GATE ARRAY AND STANDARD CELL LAYOUT

Hong Xianlong

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

Huang Jin

(Chinese Electronics Science and Technology University, Chengdu 610054)

Abstract In this paper, an efficient feed—through assignment algorithm, FARM, is proposed to minimize the maximum channel density and at the same time to reduce the wire length and via number. It connects the global routing with the channel routing. FARM consists of two parts: multi—row feed—through assignment and single row feed—through assignment. It is implemented in C language on DEC station 3100 and Sun—sparc2 station, and has been used in the Double—Layer CMOS gate array layout system—MALS3. Experiments on several benchmarks demonstrate FARM achieves comparable or better results comparing with TimberWolf 5.6.

Key words Feed—through assignment, global routing, layout for integrated circuits, gate array, standard cell.