第12卷 第2期 电子科学学刊 Vol. 12, No. 2
1990 年 3 月 JOURNAL OF ELECTRONICS Mar., 1990

LSI/VLSI 的任意元胞 K 行安置的算法*

陆生勋 姜国均
(杭州大学物理系,杭州)

摘要 本文提出一种适用于 LSI/VLSI 任意元胞布局的K行安置的算法. 当矩形单元的拓朴位置确定后,每个单元有横放、竖放两个态共有 2^n 个态. 在K行安置时,从这 2^n 个态中选出包络矩形面积最小的问题,可归结为求 n 个态中的包络矩形面积最小,所以是很有效的算法. 可以和结群法混合使用;在一定条件下,还可以直接用于准 BBL 布局.

关键词 BBL 布局; LSI/VLSI 布图;计算机辅助设计

1. 基本概念

在单元的形状均为矩形时,考虑到布线所需要的通道,采用单元扩展的方法预留布线通道. 如果原单元 A' 的某边有 c 个接线端,则将该边向外扩展 $\lceil c/k \rceil$ 条轨道, 根据经验,我们取 $k=2$. 和扩展边叠合的单元是扩展单元,简称单元.

设有 n 个单元 $A_1, A_2, \cdots, A_n$, 单元 A_i 长边的长度称为高度,记作 $h_i(A_i)$; 短边的长度称为宽度, 记作 $w_i(A_i)$, 有 $h_i(A_i) \geqslant w_i(A_i)$, $1 \leqslant i \leqslant n$. 有时简记作 h_i 和 w_i. 将单元 A_i 作图 1 所示的安置, 其中 A_i 长边的方位和 Y 轴一致,称 A_i 为竖单元, 记作 $A_i(h_i, w_i)$ 或 $\dot{A}_i$; 另一单元 A_j 称为横单元,记作 $A_j(w_j, h_j)$ 或 $\overline{A}_j$. 单元 A_i 和 A_j 的这样放置分别称为竖放和横放. 约定 $h_i = w_i$ 的单元属横放. 和 X 轴 (Y 轴) 平行的线段的长度称为 X 长度 (Y 长度).

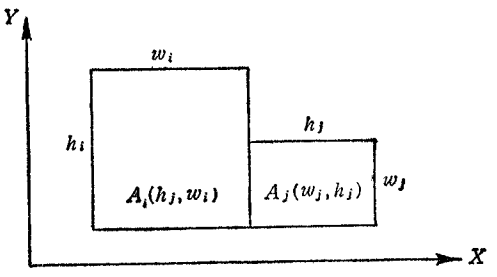

图 1 单元的安置

将单元 $A_1, A_2, \cdots, A_n$ 左右靠拢,下底对齐且平行 X 轴, 这样安置称为单行安置, 写作 $L\{A_1, A_2, \cdots, A_n\}$. 若有 k 个单行安置,将它们每个包络矩形上下靠拢, 左边对齐且平行 Y 轴,就是 K 行安置,用下式表示:

$$\begin{bmatrix} L_1\{A_{11}, A_{12}, \cdots, A_{1n_1}\} \\ L_2\{A_{21}, A_{22}, \cdots, A_{2n_2}\} \\ \vdots \\ L_k\{A_{k1}, A_{k2}, \cdots, A_{kn_k}\} \end{bmatrix} \tag{1}$$

式中 $n_1, n_2, \cdots, n_k$ 之和等于单元总数 n. 若将单行安置的单元按其高度从左到右作递

* 1988 年 4 月 25 日收到. 1989 年 3 月 23 日修改定稿.

减排列，则称这样的安置为单行有序安置，写作 $L(A_1, A_2, \cdots, A_n)$，圆括号内的单元表示有序列。将 K 行安置的每一行改为单行有序安置后就是 K 行有序安置。从每个 L_i $(1 \leqslant i \leqslant k)$ 中取一个 Y 长度最大的单元构成的集合称为墙。墙左边的单元称为墙内单元，墙右边的单元称为墙外单元。

2. 算法

K 行安置的包络矩形面积最小时称为 K 行最佳安置。求这种安置的算法分以下 5 步：

第 1 步　首先将 (1) 式所有单元（除方单元外）都竖放，记录它的包络矩形面积 S_0。从第 1 行到第 k 行以该行各单元的高度为元素作 k 个递减序列 $(h_{i1}, h_{i2}, \cdots, h_{in_i})$，这里 $h_{i1} \geqslant h_{i2} \geqslant \cdots \geqslant h_{in_i}$，$k \geqslant i \geqslant 1$。

第 2 步　处理存在竖单元的各行。作集合

$$W_u = \{\dot{A}_{ij} / 1 \leqslant i \leqslant k, 1 \leqslant j \leqslant n_i\} \tag{2}$$

式中，$\dot{A}_{ij}$ 是第 i 行中 Y 长度最大的竖单元。W 的下标 u 是集合的编号，从 1 到 n。比较一下 W_u 的 $\dot{A}_{ij}$ 那一个横放后包络面积最小就将这个 $\dot{A}_{ij}$ 横放，其他单元维持竖放，并记录这时的包络面积 S_u。

第 3 步　检查各行 Y 长度最大的是否竖单元。若是，转入第 2 步。否则，对第 i 行标志“处理完毕”，转入第 4 步。

第 4 步　检查所有各行是否都处理完毕。是就转入第 5 步。否则，转入第 2 步。

第 5 步　比较各次记录的包络面积 S_u，$0 \leqslant u \leqslant n$。设

$$S_{\min} = \mathrm{M_{in}}\{S_0, S_1, \cdots, S_n\}, \tag{3}$$

则产生 $S_{\min}$ 的 K 行安置就是最佳安置。

现在举例说明以上算法。设有 3 行安置：

$$\begin{bmatrix} L_1\{A_{11}(120,108),\ A_{12}(88,95),\ A_{13}(36,69),\ A_{14}(145,85),\ A_{15}(93,67)\} \\ L_2\{A_{21}(128,74),\ A_{22}(88,144),\ A_{23}(111,85),\ A_{24}(96,83),\ A_{25}(94,106)\} \\ L_3\{A_{31}(85,48),\ A_{32}(120,85),\ A_{33}(74,90),\ A_{34}(83,120),\ A_{35}(84,97)\} \end{bmatrix} \tag{4}$$

第 1 步　作 3 行安置如图 2 所示。包络矩形面积 $S_0 = 173416$。列出各行依单元高度递减的序列：

$(145(A_{14}), \cdots, 69(A_{13}))$,

$(144(A_{22}), \cdots, 96(A_{24}))$,

$(120(A_{34}), \cdots, 85(A_{31}))$.

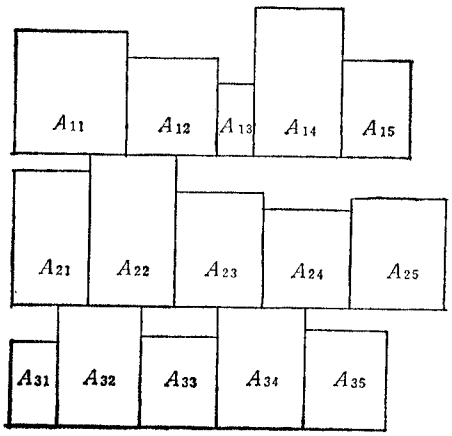

图 2　3 行安置

第 2 步　作集合 $W_1 = \{\dot{A}_{14}, \dot{A}_{22}, \dot{A}_{34}\}$。因竖单元 $\dot{A}_{14}$ 横放时包络矩形面积最小，所以将 $\dot{A}_{14}$ 横放，竖单元 $\dot{A}_{22}$、$\dot{A}_{34}$ 仍旧竖放，并记录 $S_1 = 170496$。

第 3 步　因为第 1, 2, 3 行 Y 长度最大的均是竖单元，所以转到第 2 步。作集合 $W_2 = \{\dot{A}_{11}, \dot{A}_{22}, \dot{A}_{34}\}$，比较后知道 $\dot{A}_{34}$ 应横放，记录 $S_2 = 170496$。如此反复运算，直到 3 行均处理完毕后转入第 5 步。

第 5 步　比较各次记录的包络矩形面积 S_i，知道 $\dot{A}_{14}$、$\dot{A}_{32}$ 和 $\dot{A}_{34}$ 横放，其他单元均竖放，包络矩形面积 S_3 最小，$S_3 = 161006$.

3. 算法的证明

引理 1　交换 K 行安置任意一行的各单元不会影响 K 行安置的包络矩形面积.

设有 K 行有序安置:

$$\begin{bmatrix} L_1(A_{11}, A_{12}, \cdots, A_{1n_1}) \\ L_2(A_{21}, A_{22}, \cdots, A_{2n_2}) \\ \vdots \\ L_k(A_{k1}, A_{k2}, \cdots, A_{kn_k}) \end{bmatrix} \tag{5}$$

若某行 i 从 A_{i1} 到 A_{ik} $(1 \leqslant k \leqslant in_i)$ 的单元都横放，从 A_{ik+1} 到 A_{in_i} 的单元都竖放，则称这行是正规安置行，否则，至少存在一个

$$h_{ij}(\dot{A}_{ij}) > h_{ik}(\bar{A}_{ik}), \quad k > j \tag{6}$$

且 A_{ik} 不是方单元时，称为行为非正规安置行.

引理 2　若某单行有序安置是非正规安置行，则不是单行最佳安置.

现在用归纳法证明算法的正确性. 为了便于叙述，首先根据引理 1，将 K 行安置改为 K 行有序安置如 (5) 式所示，并将所有单元(除方单元外)均竖放，记录包络矩形面积 S_0. 其次，将 (2) 式的集合 W_u 视为墙，每执行一次第 2 步的运算后放倒一个竖单元 $\dot{A}_{ij}$，将墙 W_u 的 A_{ij} 换以 A_{ij+1}，构成墙 W_{u+1}，使墙内增加一个横单元 $\bar{A}_{ij}$. 显然，记录包络矩形的面积 S_u 也就是记录墙内单元均横放，墙和墙外单元均竖放时的包络矩形面积. 若允许墙内所有 u 个单元都是自由的，可以横放或竖放，墙和墙外的单元恒竖放，则共有 2^u 个态. 当算法第 2 步运行了 u 次后，共记录了 u 个包络矩形面积 $S_1, S_2, \cdots, S_u$. 如果能证明对这 2^u 个态的最小包络矩形面积 $S_{\min}(2^u)$ 有以下关系:

$$S_{\min}(2^u) \in \{S_0, S_1, \cdots, S_u\} \tag{7}$$

则当墙内单元增加到 n 时，产生 $S_{\min}(2^u)$ 的安置就是 K 行最佳安置.

为此，先证明 $u = 1$ 时 (7) 式成立. 设 A_{i1} 是从墙 $W_0 = \{A_{11}, A_{21}, \cdots, A_{k1}\}$ 选出来的第 1 个应横放的单元. 若只允许 A_{i1} 是自由的，其他单元均竖放，则有两个态.

(1) 含 $\dot{A}_{i1}$　包络面积 S_0 在第 1 步中已记录.

(2) 含 $\bar{A}_{i1}$　包络矩形面积 S_1，现在记录. 然后将墙 W_0 的 A_{i1} 换以 A_{i2}，构成墙 W_1，使墙内增加一个横单元 $\bar{A}_{i1}$. 证得 (7) 式对 $u = 1$ 时成立.

$$S_{\min}(2^1) \in \{S_0, S_1\}$$

其次，讨论 $u = 2$ 的情况. 从墙 W_1 中选出应横放的单元 A_{jk} ($j = 1$ 时，$k = 2$，$j \neq i$ 时，$k = 1$). 若只允许 A_{i1}、A_{jk} 是自由的，其他单元均竖放，则有以下 4 个态.

(1) 含 $\dot{A}_{i1}$, $\dot{A}_{jk}$　包络矩形面积 S_0，在第 1 步中已记录.

(2) 含 $\dot{A}_{i1}$, $\bar{A}_{jk}$　(a). 若 $j = i$，则根据引理 2，可忽略这种态的包络矩形面积. (b). 若 $j \neq i$，则可以看作 A_{jk} 属于 W_0 而未被选中，它所产生的包络矩形面积大于 S_1，可以忽略.

(3) 含 $\bar{A}_{i1}$, $\dot{A}_{jk}$　包络矩形面积 S_1，已记录.

(4) 含 $\bar{A}_{i1}$, $\bar{A}_{jk}$　包络矩形面积 S_2，现在记录.

然后将墙 W_1 的 A_{i1} 换以 A_{ik}，构成墙 W_2，使墙内含有两个单元 $\bar{A}_{i1}$，$\bar{A}_{ik}$。证得 (7) 式对 $u=2$ 时成立.

现在假设墙内有 u 个单元，产生 2^u 个态，已记录 $\{S_0,S_1,\cdots,S_u\}$，对于 $u=1,2,\cdots,u$，下式成立:

$$S_{\min}(2^u)\in\{S_0,S_1,\cdots,S_u\} \tag{8}$$

以下证明 u 换以 $u+1$ 时 (8) 式也成立. 设墙 W_u 中选出应横放的单元是 A_{xy}.

(1) 若墙内所有 u 个单元均横放，墙外单元均竖放，单元 A_{xy} 若是自由的，则有两个态. 含 $\dot{A}_{xy}$ 的包络矩形面积 S_u 已记录；含 $\bar{A}_{xy}$ 的包络矩形面积 S_{u+1} 现在记录.

(2) 若墙内 W_i $(1<i<u)$ 内所有 i 个单元均横放，墙外单元均竖放，单元 A_{xy} 是自由的，则有两个态. (a) 含 $\dot{A}_{xy}$: 可以看作属于 W_i 墙内单元均横放，其余单元均竖放的情况，包络面积 S_i 已记录. (b) 含 $\bar{A}_{xy}$: 此时的包络矩形面积设为 S_w. 若允许 W_i 墙内单元是自由的，墙和墙外单元恒竖放，则有 2^i 个态. 今从 $W_i\cap\{A_{x1},A_{x2},\cdots A_{xy-1}\}$ 中任取一个单元 A_{xj} 将它竖放，其余 W_i 墙内单元仍竖放，显然这是 2^i 个态中的一个态. 设其包络矩形面积为 S_x，根据 (8) 式

$$S_x\geqslant S_{\min}(2^i)=\mathrm{Min}\{S_0,S_1,\cdots,S_i\} \tag{9}$$

另一方面，因 $\dot{A}_{xj}$ 和 $\bar{A}_{xy}$ 在同一行，根据引理 2，$S_w\geqslant S_x$. 由此可见，S_w 或者属于 $\{S_0,S_1,\cdots,S_i\}$，已记录，或者可以忽略. 综上所述，并将墙 W_u 的 A_{xy} 换以 A_{xy+1}，构成墙 W_{u+1}，即证得 $u+1$ 时 (8) 式也成立，因此算法正确.

本算法已编程通过，可以和结群法混合使用[1]，并结合最小割法用于准 BBL 初始布局.

参　考　文　献

1] 俞明永，陆生勋，庄文君，全国第四届集成电路 CAD 学术年会论文集，哈尔滨，1987 年.

AN ALGORITHM OF *K*-LINE LOCATION FOR BBL IN LSI/VLSI

Lu Shengxun　Jiang Guojun

(Hangzhou University, Hangzhou)

Abstract A *K*-line location algorithm for buillding block cells in LSI/VLSI is presented. When the relative positions of rectangular cells are given, there are 2^n states according to the two orientations of a cell. It is proved that to find the optimum solution from 2^n states can be reduced to calculate the n states in k-line algorithm. So the algorithm is very effective and can be used with association for cluster method in BBL placement. Under certain conditions, this method can also be used to pesudo BBL placement directly.

Key words　BBL placement; Layout of LSI/VLSI; CAD