第5卷 第4期 半 导 体 学 报 Vol. 5, No. 4
1984 年 7 月 CHINESE JOURNAL OF SEMICONDUCTORS July, 1984

# LSI 多元胞版图设计的一种布局方法

沙 露 唐 璞 山
(复旦大学电子工程系)

1983 年 7 月 20 日收到

本文提出 LSI 版图自动布局的一个实用程序. 该程序适用于多元胞布局模式,由块的划分和块内布局两个步骤完成. 求解的算法主要有: 分配问题的近似算法、Kernighan-Lin 算法和分枝界限法. 文中着重介绍对上述算法的改进. 主要的改进为: 1, 在采用分配问题求解时, 定义了三个选择函数以适应不同的要求, 其中 $F_2$ 带有一定预见性; 2, 对 Kernighan-Lin 算法的目标函数及交换方式作了改进, 以满足布局问题划分中特有的几何参数的要求. 此外在迭代过程中使用了分枝界限法,通常可显著减少迭代时间. 该程序使用 Fortran-IV 语言编写,已在 PDP-11/34 计算机上调试通过,并运行了若干实例.

## 一、引 言

目前用设计自动化技术来设计 LSI 版图的步骤是: 先建立基本逻辑单元(如门、触发器或功能器件)的版图数据库,在版图设计时调用版图数据库中的单元, 先用布局程序确定单元在版图上的位置和方向,再用布线程序实现单元之间的电学互连.

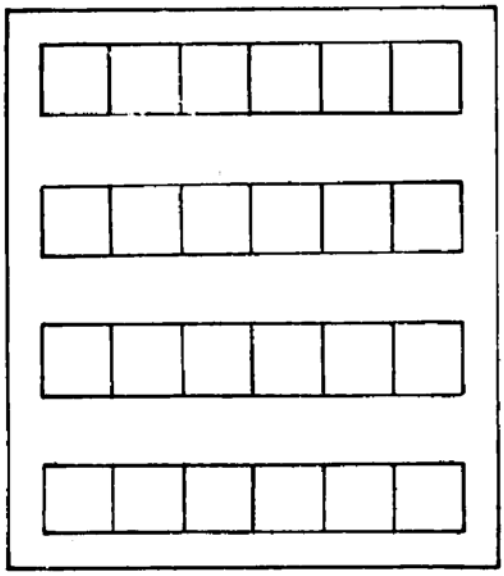
为了更好地利用设计自动化技术, 对版图单元的几何参数及其在版图上的排列方式做不同的规定,这就形成了不同的布局模式. 目前国际上常用的布局模式有三种,如图 1 所示. 图 1(a) 的图案称为母片式 (Masterslice) 版图[1], 它所使用的单元几何参数完全相同,这些单元在版图上排成阵列,称作母片, 一种母片对于同类规模不同品种的电路是通用的,仅是铝布线版不同;图 1(b) 的图案称为多元胞式 (Polycell) 版图[2], 它所使用的单元宽度必须相同而长度可不同,单元在版图上分行排列,不同电路使用的单元品种和数量不同;图 1(c) 的图案称为任意元胞式 (Arbitrary Cell) 版图[3], 它的版图单元通常是任意长宽的矩形,排列方式也不作限制,因此接近于人工设计的版图. 本文提出一种适用于多元胞式版图设计的自动布局程序.

## 二、多元胞式版图的布局

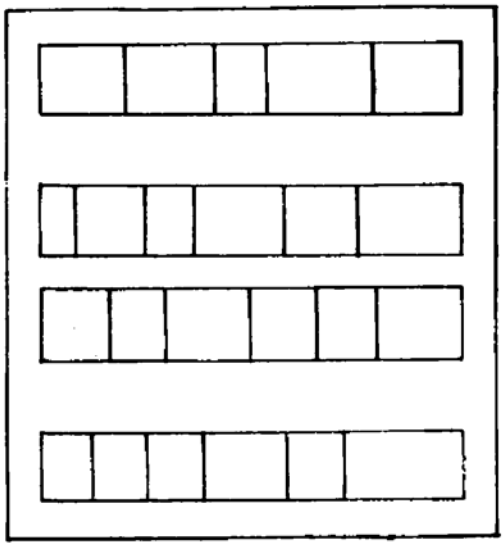
### I. 基本规则

1. 元胞 (Cell): 版图中的基本单元称为元胞. 本文中对所用元胞(如图 2 所示)有以下限制: 同一版图所用元胞的宽度 ($W$) 相同而长度 ($L$) 可不同; 元胞的逻辑功能管脚单侧均匀排列(可以有空脚)*;电源线和地线引出脚位于元胞的左右两侧,这对 CMOS 电

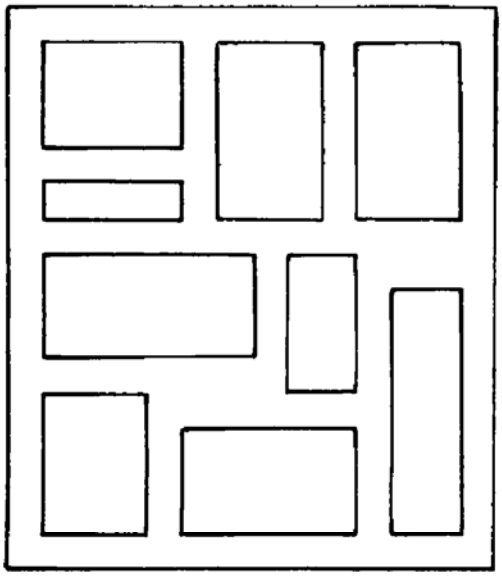
* 最近对程序稍作改进,适合有两侧引出脚的元胞,块间也有水平通道作块间连线之用, 即块间连线不再利用两侧垂直通道.



(a) 母片式版图



(b) 多元胞式版图



(c) 任意元胞式版图

图 1

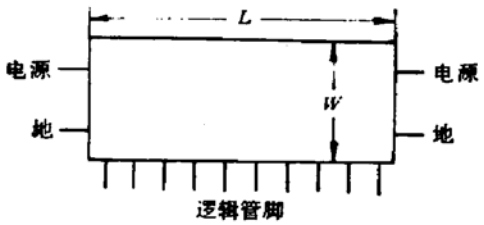



图 2

路版图的设计特别适合.

2. 块 (Block): 如图 1(b) 所示,逻辑功能管脚相对的两行元胞组成一个块.

3. 通道 (Channel): 如图 1(b) 所示,块中两行元胞所夹的空间叫作水平通道,块左右两侧的空间叫作垂直通道，通道作为互连布线的区域. 位于同一块中的元胞之间的互连

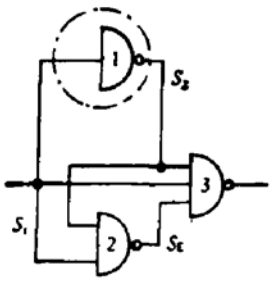
线布在该块中的水平通道上；位于不同块中的元胞之间的互连线经过两侧的垂直通道.

### II. 布局设计

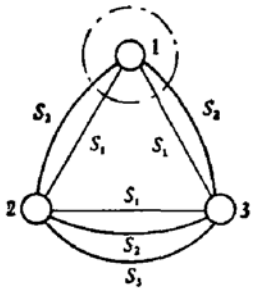
我们把在平面上布局的二维问题分解为两个一维布局问题来处理. 首先完成 $y$ 方向上的布局，即划分块；然后完成 $x$ 方向上的布局，即块内布局. 完成二步后即可确定元胞在版图上的位置.

## 三、电路描述模型

元胞之间的互连使各元胞的最优位置互相制约是布局问题复杂化的重要原因. 多数算法[4,5,6]把元胞之间的互连关系简化为用线图及其对应的邻接矩阵来表示，有时把布局问题转化为二次分配问题[7]求解. 这种转化实质上是把位于同一个信号组中的 $S$ 个元胞之间的互连形式由最小生成树[7]简化为分别考虑这些元胞之间的两两互连. 当 $S>2$ 时就会引入偏差，且 $S$ 越大偏差也越大. 在图 3 中，(a) 是一个电路的逻辑图；(b) 是与 (a) 相应的二次分配图及对应的邻接矩阵，从图中很容易看到把元胞 1 划分出来两者所需的切割线不同，即引入了偏差. 考虑到二次分配问题虽然比布局问题有所简化，但仍然是一个 $NP$ 完全问题，即仍无求最优解的有效算法. 本文不采用二次分配图模型而是采用偶

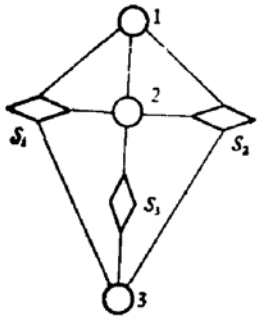



(a) 逻辑图




| 元胞 \ 元胞 | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 0 | 2 | 2 |
| 2 | 2 | 0 | 3 |
| 3 | 2 | 3 | 0 |

(b) 二次分配图及邻接矩阵




| 元胞 \ 元胞 | 1 | 2 | 3 |
|---|---|---|---|
| 1 | 1 | 1 | 0 |
| 2 | 1 | 1 | 1 |
| 3 | 1 | 1 | 1 |

(c) 偶图及关联矩阵

图 3

图模型[8]来描述元胞之间的互连关系. 图 3(c) 是该逻辑图的偶图模型，图中有两类顶点分别代表信号组和元胞，同类顶点之间互不连接. 在与偶图对应的关联矩阵中，行对应元胞；列对应信号组. 矩阵中的么元 $c_{ij}=1$ 表示信号组 $j$ 中包含元胞 $i$. 偶图中把元胞 1

划分出来所需切割线的数目与原逻辑图完全一样。

关联矩阵采用稀疏矩阵的存放方式，由于该矩阵只有零元和么元，故只需记录其么元的位置而不必记录数值．对多数电路存放其关联矩阵比存放邻接矩阵节省内存．偶图模型较之二次分配图模型的不方便之处是计算元胞之间的联结度时查找次数较多，因而增加了计算时间，为此将关联矩阵按行指针和列指针分别存放两套数据，这虽增加了信息的冗余度，但可使查找一个信号组中包含那些元胞的操作次数与电路的规模（元胞总数 $n$）无关，而仅与该信号组中包含元胞个数（通常为 2—5）有关，故不致使计算时间显著增加．

## 四、划分块的算法

划分块的任务是把所使用的全部元胞的集合划分为有序排列的若干子集（块），使得各块中的元胞面积之和大致相等，且各块间的互连线总长度最小．

逻辑图采用偶图描述之后，划分块可看作偶图的划分问题，但约束条件和目标函数都需要考虑元胞的几何参数．由于各元胞的面积不尽相同，故不能将元胞简单地看作偶图中的一个顶点，这就增加了分块问题的复杂性，使得针对拓朴图划分的一些理论上看来较好的算法，在实用中不一定能得到较好的结果．人工设计者虽然也没有一定的规律可循以保证得到最优解，但有经验的设计者经过多次尝试后却往往可找到比自动设计更好的解，这是因为人工设计者有两个显著的长处：（1）有一定预见性，在确定一个元胞的分配时，不仅根据它与已布元胞的联系而且还考虑它布上以后对后面有什么影响；（2）较全面地看问题，每做一步都统筹兼顾各方面参数的要求．本文在选择和改进算法时，力图吸取上述两个长处，主要考虑是：（1）在初始划分的分配算法中定义了带有预见性的选择函数 $F_2$，在迭代改善算法中选用具有预见性的 Kernighan-Lin 算法[4]；（2）在算法中加进了对几何参数的考虑，使得每一步都兼顾块间连线短和块长差别小两方面的要求．

分块由构造初始划分和迭代改善二步来完成．每迭代一次产生一个新的分块方案，直至迭代停止，由用户选出较优的一个分块方案．

### I. 初始划分

初始划分的任务是：把 $n$ 个元胞分到 $m$ 块，使块间连线总数最小，且各块所含元胞的个数相差最小．我们采用构造法进行分配，即按联结度依次将元胞分入 $m$ 块．该方法的核心是计算分配费用的目标函数，即是计算元胞之间联结度的选择函数．在初始划分中定义了两个选择函数．

1. 选择函数 $F_1$

在当前已布元胞的基础上，待分配元胞 $x$ 与第 $K$ 块的联结度的定义为：

$$F_1(x, B_K) = \frac{|S(x, B_K)|}{|S(x)|}. \tag{1}$$

式中 $B_K$ 是第 $K$ 块已布元胞的集合；$S(x)$ 是含有元胞 $x$ 的信号组集合；$S(B_K)$ 是含有 $B_K$ 元胞的信号组集合；$S(x, B_K) = S(x) \cap S(B_K)$；$|S|$ 是集合 $S$ 的势，即集合 $S$ 中包含的信号组个数．$F_1$ 的物理意义是待布元胞 $x$ 与第 $K$ 块已布元胞集合公有的信号组个数与元

胞 $x$ 所连信号组总数之比，是归一化的相对联结度.

2. 选择函数 $F_2$

$F_2$ 的定义为:

$$F_2(x,B_K)=|S(x,B_K)|-|S(x,B)|-\sum_{S_i\in S(x,B_k,A)}\rho_1(S_i)+\sum_{S_i\in S(x,B,A)}\rho_2(S_i) \quad (2)$$

式中 $A$ 是除 $x$ 之外其余待分配元胞的集合; $B$ 是除第 $K$ 块之外其余块中已布元胞的集合; $S(A)$ 是含有 $A$ 元胞的信号组集合; $S(B)$ 是含有 $B$ 元胞的信号组集合;

$$S(x,B)=S(x)\cap S(B);\ S(x,B_K,A)=S(x)\cap S(B_k)\cap S(A);$$

$$S(x,B,A)=S(x)\cap S(B)\cap S(A).$$

$F_2$ 具体考虑了元胞 $x$ 所连各信号组中的元胞分配情况. $F_2$ 不仅考虑了 $x$ 与第 $K$ 的连接，还考虑了与其余块的连接; 此外还对与 $x$ 有联系的待分元胞以后能分入 $K$ 块的几率作出预测，作为 $F_2$ 的修正项，即 (2) 式中的第三、四项，其中 $\rho_1$ 和 $\rho_2$ 分别为:

$$\rho_1(S_i)=1-\left(\frac{1}{m}\right)^{n_x(S_i)}, \quad (3)$$

$$\rho_2(S_i)=1-\left(\frac{m-1}{m}\right)^{n_x(S_i)}. \quad (4)$$

式中 $m$ 是划分的块数; $n_x(S_i)$ 是信号组 $S_i$ 所包含 $A$ 集合中的元胞个数. $F_2$ 的物理意义是反映元胞 $x$ 分入第 $K$ 块后使该块引出线数目改变的一个物理量.

在初始划分过程中单独使用 $F_1$ 或 $F_2$，或 $F_1$ 和 $F_2$ 先后结合使用，可由人机对话控制. 一般应对此做几种选择以得到几个不同的初始划分方案(初始划分时间一般比迭代时间小)，再从中选一个方案进行迭代. 实例说明，$F_1$ 和 $F_2$ 结合使用得到的初始划分方案迭代后得到的结果较优.

初始划分的计算步骤如下:

(1) 输入控制从使用 $F_1$ 转为使用 $F_2$ 的比例因子 $E$;

(2) 块号指针 $K$ 置 0，分配轮次奇偶标志 $T$ 置 1;

(3) $K=K+T$;

(4) 计算本轮次已分元胞数与元胞总数之比值 $R$;

(5) 计算各待分元胞与 $K$ 块的联结度(若 $R<E$，使用 $F_1$，否则使用 $F_2$)，并记录在数组 $AC(n)$ 中;

(6) 查找 $AC(n)$ 中的最大值，将其对应的元胞序号(可能是一个，也可能是多个)记录在数组 $KM(10)$ 中;

(7) 设 $L_0$ 为本轮次已分元胞的平均块长，$L(K)$ 为 $K$ 块已分元胞的块长，若

$$L(K)<0.9L_0,$$

从 $KM(10)$ 中选长度最大的元胞分入; 若 $L(K)>1.1L_0$，选长度最小的元胞分入; 否则从 $KM(10)$ 中任取一个元胞分入;

(8) 若待分元胞个数为 0，初始划分完成; 否则 $K=K+T$，若 $0<K<m$，转 (5); 否则置 $T=-T$，转 (3).

上述步骤采用了并行划分的原则，且奇偶轮次按相反的块号顺序分配，以使各块选

择元胞的机会比较均等。

## II. 迭代算法

常用的迭代算法是成对交换法，但成对交换法每次只以一对元胞交换后的得失决定取舍，难以达到整体优化的要求．我们选用 Kernighan-Lin 算法，该算法着眼于交换若干对元胞后的累计增益，有一定预见性．本文把该算法用于布局中，做了下述改进：

1. 输入逻辑图模型

Kernighan-Lin 算法采用二次分配模型，本文则改用偶图模型。

2. 目标函数

Kernighan-Lin 算法的目标函数是块间连线数最小，即

$$C = \frac{1}{2}\left[\sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m} c(i,j)\right]. \tag{5}$$

式中 $i$ 和 $j$ 是块号；$m$ 是划分块数；$c(i,j)$ 是 $i$ 块和 $j$ 块公有信号组个数．改进后的目标函数为块间连线总长度最小，即

$$L^* = \frac{1}{2}\left[\sum_{i=1}^{m}\sum_{\substack{j=1\\ j\neq i}}^{m} c(i,j)|i-j|\right]. \tag{6}$$

式中用块序号之差 $|i-j|$ 来表示 $i$ 块和 $j$ 块之间一条连线的长度．由于 $L$ 中加了权重因子 $|i-j|$，即相隔较远的块间引线要求更少。

3. 交换方式

布局要求各块中元胞的面积相等而不要求各块元胞个数相等，为此将 Kernighan-Lin 算法所规定的二块间元胞成对交换扩展为 1 对 $n$($n \geqslant 0$，是交换轮次的函数)交换．不对等交换的条件是：(1) 不对等交换后可使二块长度差减小；(2) 不对等交换的块间连线长度增益不小于成对交换增益。

4. 多块间迭代处理

Kernighan-Lin 算法针对块数为 2 的情况，当块数大于 2 时，迭代效果与块的处理顺序有关，故我们在算法中增加了动态确定处理顺序的步骤，其原则是优先处理块长指标较差的块．

迭代算法的计算步骤如下：

(1) 根据动态确定处理顺序的原则，确定本轮次要处理的一对块 $B_i$ 和 $B_j$；

(2) 元胞交换次数指针 $K$ 置 1；

(3) 设 $L_c$ 为所用的最小元胞长度，$L(i)$ 和 $L(j)$ 分别为 $i$ 和 $j$ 块的块长，若

$$|L(i) - L(j)| < \frac{2}{3} L_c,$$

转 (6)；否则转 (4)；

(4) 查找块长较大的块中是否存在适合单方换块的元胞 $x$，如把 $x$ 换入另一块，可使 $|L(i) - L(j)|$ 减小且不增加块间连线的总长度，则将 $x$ 换入另一块，转 (5)；否则转 (6)；

(5) 记录换块的元胞序号及块间连线总长度的累计增益 $G(K)$，修改块长 $L(i)$，$L(j)$，$K-K+1$，转 (3)；

(6) 若 $B_i$、$B_j$ 二块中都存在尚未交换过的元胞，用分枝界限法从中找出交换后可获得最大增益(不一定是正增益)的一对元胞，实行交换后转 (5)；否则转 (7)；

(7) 从累计增益序列 $G(K)$ 中查找最大元 $G(P)$，$P$ 序号以后交换的元胞全部复原；

(8) 若有未处理过的块对，转 1；否则迭代完成.

## 五、块内布局算法

块内布局是把各块中的元胞分为长度接近的上、下两行，并确定各行中元胞的排列顺序，使块内水平通道中连线长度最短. 块内布局分两步进行：首先根据几何参数的要求分行；然后根据布线的要求完成行内一维布局.

### I. 分行

目标函数是使上、下两行长度之差较小. 采用的算法是将块中所有元胞按其长度递减的顺序排列，然后依次取元胞放在当前较短的一行中 (若两行等长则放在下边一行). 在分行过程中，两行长度之差始终不大于最新放入的元胞长度. 长度较小的元胞后放可使两行长度之差较小，且两行的元胞个数差别最小.

### II. 行内一维布局

目标函数是使各水平通道中的布线总长度最小. 求解的步骤与初始划分中采用的分配算法的步骤类似，一维布局所用的选择函数 $F_3$ 的定义为：

$$F_3(y)=\frac{|S(Y,B)|+|S(y,IO_L)|-|S(y,IO_R)|}{|S(y)|}. \tag{7}$$

式中 $y$ 是某块内待定顺序的元胞；$B$ 是某块内已排定行内顺序的元胞集合；$IO_L$ 位于芯片左侧的轮入输出端集合；$IO_R$ 位于芯片右侧的输入输出端集合，$S$ 的定义与 $F_1$ 中相同. 同一块中元胞排列顺序是上、下行交替进行，排列时选择该行待分元胞中 $F_3$ 值最大的元胞放在已排列元胞的右邻.

## 六、布局程序结构

布局程序的总框图如图 4 所示. 其中预处理是根据输入数据对图形面积、划分块数、平均块长等有关数据作出估算，作为后来布局的基础；硬种子是指芯片四周的引出端；软种子是指预先分入各块中的元胞，软种子的选择是否恰当对初始划分结果影响很大，本程序是由人工选择输入，选择时应注意不宜把相互之间联结度大的元胞分入不同的块作为软种子；指标评价是对现有分块方案的各项指标(包括块长差别、块间连线总长度、块间引出线数目等)给出评价，并计算现有分块方案的上述指标比前一个分块方案的改善率，作为判断迭代改善的标准.

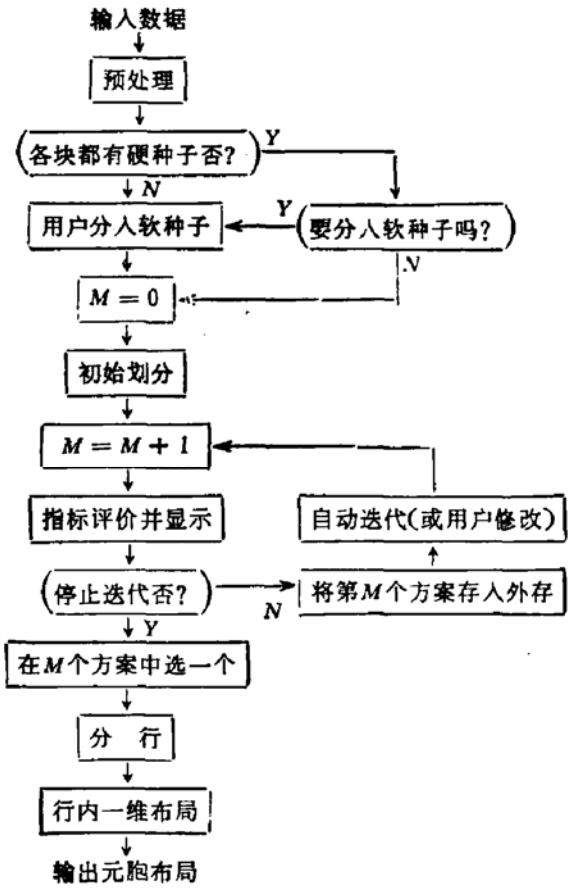



图 4

## 七、结　果

本文介绍的布局程序使用 Fortran-IV 语言编写，采用了若干节省内存的措施以适于小型机上运行，在 PDP-11/34 计算机上可以处理的电路规模不小于 500 个元胞组成的电路.

表 1 给出了例 1(18 个元胞的电路分为二块)中 $E$ 值 ($F_1$ 转换到 $F_2$ 的比例因子)不同

表 1

| 初始划分 | | | 迭代 | | |
|---|---|---|---|---|---|
| $E$ 值 | 块长 | $L$ | 次数 | 块长 | $L$ |
| 0～0.1 | 106 | 8 | 1 | 109 | 5 |
| 0.2 | 109 | 8 | 1 | 102 | 6 |
| 0.3～0.8 | 109 | 5 | 0 | 不变 | 不变 |
| 0.9～1.0 | 105 | 7 | 1 | 109 | 5 |

注：表中迭代次数是到再迭代下去指标无改善为止的次数.

对初始划分的影响，结果证明 $E$ = 0.3—0.8 的结果最好，初始划分就已达到最优结果（本例中块间连线长度 $L$ = 5 是最佳，不需再迭代改善）.

表 2

| 例 | I/O 端个数 | 元胞个数 | 信号组个数 | 划分块数 | 初始划分时间（秒） | 迭代一次时间（秒） | 指标评价时间（秒） | 块内布局时间（秒） | 数据区所占内存量（字） |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 14 | 18 | 22 | 2 | 1.52 | 1.86 | 2.34 | 3.36 | <400 |
| 2 | 8 | 24 | 28 | 2 | 2.14 | 2.01 | 2.56 | 4.04 | <480 |
| 3 | 14 | 64 | 73 | 3 | 12.70 | 18.9 | 3.20 | 17.80 | <1000 |
| 4 | 58 | 126 | 158 | 4 | 77.51 | 157.0 | 3.22 | 192.76 | <2500 |

表 2 给出了 4 个电路实例的规模、划分情况、运行时间及数据区所占的内存量. 从已运行的实例看，使用该程序可以得到令人满意的结果.

本程序作为复旦 CAD 实验室多元胞芯片设计系统的一部分，该设计系统已开始用来设计硅栅 CMOS 电路. 布局结果除直接显示外，尚形成一个磁盘数据文件供其后的布线程序作为初始数据调用.

作者感谢陆昉同志编了部分预处理程序、吴祖增同志给予帮助以及 PDP-11/34 机房同志的支持.

## 参 考 文 献

[1] S. Goto, 16th Design Auto. Conf. Proc., p. 11 (1979).
[2] G. Persky, D. N. Deutsch and D. G. Schweikert 13th Design Auto. Conf. Proc., p. 399 (1976).
[3] B. T. Preas and W. M. VanCleempt 16th Design Auto. Conf. Proc., p. 474 (1979).
[4] B. W. Kernighan and S. Lin, *Bell System Tech. J.* **49**, 291 (1970).
[5] L. Steinberg, *SIAM Rev.* **3**, 37 (1961).
[6] N. R. Quinn Jr. and M. A. Breuer, *IEEE Tran. on Circuits and System*, CAS-26 377 (1979).
[7] M. A. 布鲁尔主编[美]"数字计算机设计自动化的理论和方法"，科学出版社，1978年出版.
[8] 可儿贤二等[日]"计算机辅助设计——LSI" p. 92，科技文献出版社重庆分社1979年出版

# An Approach to Polycell Placement for LSI

Sha Lu and Tang Pushan

*(Dept. of Electronics Engineering, Fudan University)*

## Abstract

This paper presents a placement technique devoted to polycell layout of LSI and involving two steps: partitioning a circuit into blocks and placing the cells in them. The main algorithms used are: approximation method for assignment problem, Kernighan-Lin's method and branch-bound method. Improvements for these algorithms are proposed: 1. For solving assignment three selective functions are defined and among which $F_3$ is with prediction; 2. The objective functions and the interchange form of the Kernighan-Lin's method are tailored to meet the special requirements of geometric parameters in partitioning. In addition, the branch-bound method is used in iterative process, these making it possible to reduce the search time in many situations.

The schematic diagram is given. The program is written in Fortran-4 language and has been implemented on PDP-11/34 computer.