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

可编逻辑阵列的折迭和版图自动生成

赵 文 庆　　唐 璞 山
（复旦大学电子工程系）

1983 年 12 月 5 日收到

本文提出了一个可编逻辑阵列（PLA）的自动设计系统。它以逻辑函数作为输入数据，先后进行逻辑综合、PLA 折迭以及自动生成版图的制版源程序。适用于多输出函数组合逻辑电路的设计，能够处理的输入变量和输出变量总数最多为 32。文中对系统的两个主要部分作了介绍。首先，叙述了一种改进的折迭算法，提出了用以提高折迭效率的折迭参数法和交叉折迭法，并介绍了完整的折迭过程。其次，叙述了版图自动生成的方法以及按照双极型工艺实现版图生成的过程。最后给出了本系统运行的若干结果。

一、引　言

可编逻辑阵列——FGPLA，是适合于实现多输出函数组合逻辑的阵列形式的电路。传统的 PLA 由一个与阵列和一个或阵列相联而成(图 1)。它的致命弱点是稀疏性，典型的有效密度在与阵列和或阵列中分别为 10% 和 4%。近十年来，许多人对 PLA 面积的优化作了研究，发展了各种各样的折迭技术。将与阵列和或阵列分别拆成若干个阵列，将输入线、输出线和积项线分别折断，由单侧输入、单侧输出改为多向输入和多向输出，从而获得高密度的 PLA (图 2)[1-3]。

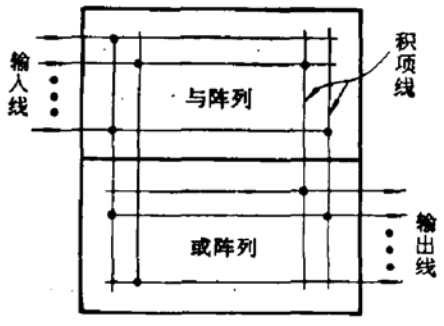

图 1　传统的 PLA

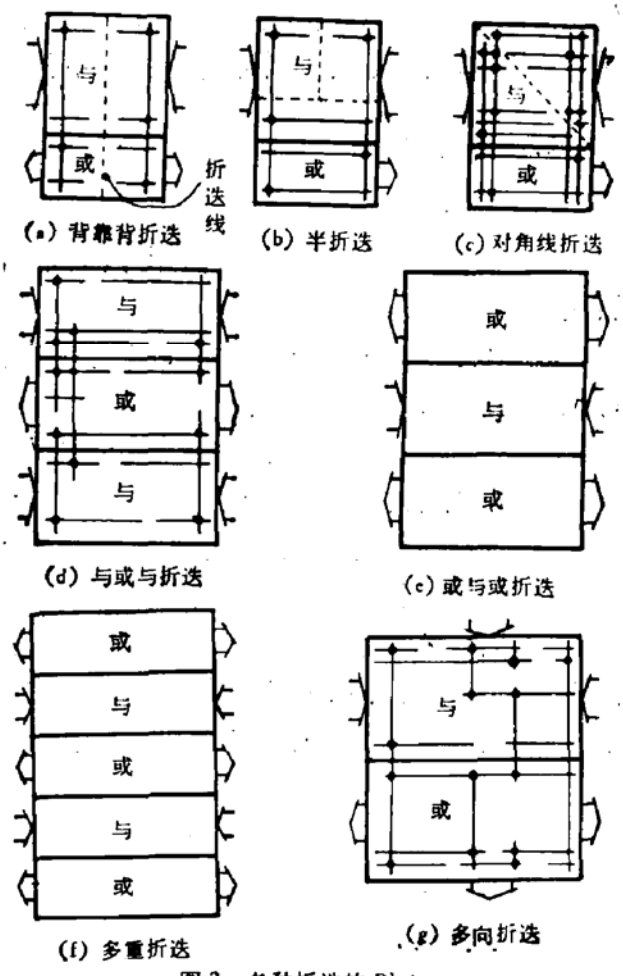

图 2　各种折迭的 PLA

根据 PLA 规则性强的特点，本文提出了一个 PLA 自动设计系统 FGPLA (Folding and Mask Generation of PLA) 它以逻辑函数作为输入数据，在逻辑综合的基础上[4]，采用与或与的阵列形式，先将稀疏的阵列进行折迭，然后根据符号版图，参照一定的工艺参数和设计规则，自动生成制版语言的源程序. FGPLA 的这一结果可以利用集成电路辅助制版系统[5]制出掩膜版. 它既可以作为大芯片内部的子电路，也可以加上外围电路后单独使用.

文中分别对 FGPLA 的两个主要部分作了介绍. 首先，提出了一种新的折迭算法，讨论了用折迭参数法和交叉折迭法来提高折迭效率的问题，并简述了实现输入线、输出线和积项线三个方向折迭的过程. 其次，叙述了如何根据设计规则和工艺条件来自动生成双极型版图的设计过程.

FGPLA 是我系研制的大规模集成电路计算机辅助设计系统的一部分，可用于输入变量和输出变量总数不超过 32 的 PLA 的设计. 采用 FORTRAN IV-PLUS 语言编制了程序，在 PDP-11/34 计算机系统中调试和运行，获得了预期的结果.

二、PLA 折 迭

PLA 折迭是通过重新安排阵列的行和列来提高阵列的有效密度. 对于仅仅考虑列的优化折迭问题——简单列折迭 SCF，求它的最优解是一个 NP 完全性问题. G. D. Hachtel 等人在 1982 年提出了一个有效的折迭算法模型[6]，本算法就是在此基础上发展的.

1. 折迭算法

(1) 定义

设 $P=[p_{ij}]$ 是 PLA 的符号矩阵，r_i 和 c_j 是它的行和列，当 r_i 和 c_j 相交时，$p_{ij}=1$，否则 $p_{ij}=0$.

设 $R(c_i)=\{r_1^i,\cdots,r_k^i\}$ 是 c_i 列中非零行的集合. 若 $R(c_i)\cap R(c_j)=\phi$，表示 c_i 和 c_j 是可折迭的. 并定义关系

$$Q_l=R(c_i)<R(c_j)=\{r_m^i<r_n^j \mid r_m^i\in R(c_i), r_n^j\in R(c_j)\}$$

表示 $R(c_i)$ 中各行都必须位于 $R(c_j)$ 中各行之上，称 $o_l=(c_i,c_j)$ 为由 Q_l 所确定的有序折迭对. 设 $0=\{o_l\}$ 是有序折迭对的集合，对应的关系集合 $\{Q_l\}$ 的并集记为 $Q(0)$.

设 R 是一关系集合，对于其中任意元素 r_i、r_j 和 r_k，如果满足：i) 自反关系，即存在 $r_i<r_i$；ii) 传递关系，即若存在 $r_i<r_j$ 和 $r_j<r_k$，必定存在 $r_i<r_k$；iii) 反对称关系，即若存在 $r_i<r_j, i\neq j$，则必不存在 $r_j<r_i$；则称 R 为偏序关系集. 并把对称关系(存在 $r_i<r_j, r_j<r_i$ 且 $i\neq j$) 称为折迭中的循环约束.

设 R 和 R' 是两个关系集，且有 $R\subseteq R'$，如果满足当存在

$$(R_l<r_m)\in R,\ (r_m<r_n)\in R,$$

而必有 $(r_l<r_n)\in R'$ 时，称 R' 是 R 的传递扩张. 并把 $Q(0)$ 的所有继续传递扩张的并集称为 $Q(0)$ 的传递闭包，记为 $Q^+(0)$.

(2) 折迭的可实现性定理

设 O 是符号矩阵 P 的有序折迭集合，当且仅当 $Q(0)$ 的传递闭包 $Q^+(0)$ 加上自反关系后是 R 上的偏序关系集时，有序折迭集合 O 是可实现的.

根据 $Q^+(0)$ 的偏序关系，可以得到 P 中各行 r_i 的排列次序，由于反对称的性质，不会出现循环约束，因而折迭是可实现的. 反之，若出现对称关系 $(r_i < r_j)$ 和 $(r_j < r_i)$，即为循环约束，因而有序折迭集合 O 是不可实现的.

根据这一定理，折迭的优化问题就等价为寻找最大的可实现的有序折迭集合的图论问题. 然而，这仍旧是一个 NP 完全性问题，为此我们引进以下的启发式算法.

(3) *折迭图和折迭算法*

符号矩阵中各列的可折迭情况可用折迭图 $G=(V, E)$ 来表示. 其中 V 是顶点集合，与符号矩阵的列一一对应. E 是可折迭列的边集，

$$E=\{e=(v_i, v_j) \mid R(c_i) \cap R(c_j)=\phi\}.$$

称顶点所关联的边数为它的折迭度，$R(c_i)$ 中非零行的数目为顶点的价格. 设 M 是 G 的对集，如果 M 是可折迭的，则称为折迭对集 M^*. 对某一折迭对集 M^*，如果不存在任何折迭对集 M，使得 $|M^*| < |M|$，则称 M^* 为最大折迭对集.

在以下的启发式算法中，将根据顶点的折迭度和价格，先难后易地逐步选择顶点对，构造并扩大折迭对集 M，使它始终是可实现的，并在折迭过程结束时，得到近似的最大折迭对集 M^*，从而获得有序折迭集合 O.

折迭算法如下：

```
begin M = 0; TRANS = φ;
11: while (V≠φ) do
    begin v = VSELECT(V); U(v) = {u ∈ V|(v, u) ∈ E};
12: if (U(v) = φ) then VDELET (v); goto 11;
      else begin u = USELECT(U(v)); CLOS = CLOSVU (v, u, TRANS);
           if (AC(CLOS)) then
             begin CLOS = CLOSVU (u, v, TRANS);
               if (AC(CLOS)) then UDELET (u); goto 12;
                 else M = M∪{(u, v)};
             end;
             else M = M∪{(v, u)};
           VDELET (v, u); TRANS = CLOS;
         end;
    end;
end;
```

其中 TRANS 是传递闭包，CLOS 是临时的传递闭包. $U(v)$ 是顶点 v 的折迭对象集合. VSELECT、USELECT、CLOSVU、AC、VDELET 和 UDELET 都是子程序，功能如下：

1) VSELECT 和 USELECT 分别是在顶点集合 V 和 $U(v)$ 中选择折迭边的起点 v 和终点 u 的子程序. 选择原则是优先选中折迭度最小，价格最大的顶点.

2) CLOSVU 是构造传递闭包的子程序. 传递闭包表示了行与行之间的关系，可以用邻接矩阵 $A=[a_{ij}]$ 来表示，当存在关系 $r_i < r_j$ 时，$a_{ij}=1$，否则 $a_{ij}=0$. 对于关系 $r_i < r_j$，称 r_i 是 r_j 的祖先，r_j 是 r_i 的后代. 构造传递闭包是把折迭对所对应的所有祖先和后代的关系添加到原有的传递闭包中去并进行传递扩张.

邻接矩阵 A 的存贮结构将决定数据的存贮量和 CLOSVU 的计算量，从而决定折迭算

法的计算量．设符号矩阵 P 的行数为 R，则 A 的容量为 R^2． 现在采用按位存贮的方法，使用一个二维数组 CLOS，元素个数为 $m \cdot R$，其中 $m = \left[\frac{R-1}{W}\right] + 1$[注]，$W$ 是计算机的字长． CLOSVU 的基本运算是用按位逻辑加的方法实现传递闭包中某两行之间的并运算，运算次数为 m． 假定 $N_v = |V|$，在最坏情况下，CLOSVU 的调用次数为

$$N_v(N_v - 1)/2,$$

每次调用时最多的运算次数为 mN_v^2，所以，CLOSVU 的计算量是 $O(mN_v^4)$．

3) AC 是用来检查 CLOS 是否存在反对称关系的．根据传递闭包的构造过程，可折迭对集 M^* 的判据是传递闭包中不存在对称关系，如果存在对称关系，由于传递，必定会出现自反关系．因而在执行子程序 AC 时，一旦发现某个对角元素不等于零时，表示存在对称关系，AC 就给出肯定的结果．

4) UDELET 是从 $U(v)$ 中删去顶点 u 的子程序． VDELET 不仅从 V 中删去顶点 v，还要：

i) 删去与 v 相联的边，$E = E - \{(v,v')|(v,v') \in E\}$，并使所有 v' 的折迭度递减；ii) 删去由于 i) 而使折迭度减为零的所有顶点．

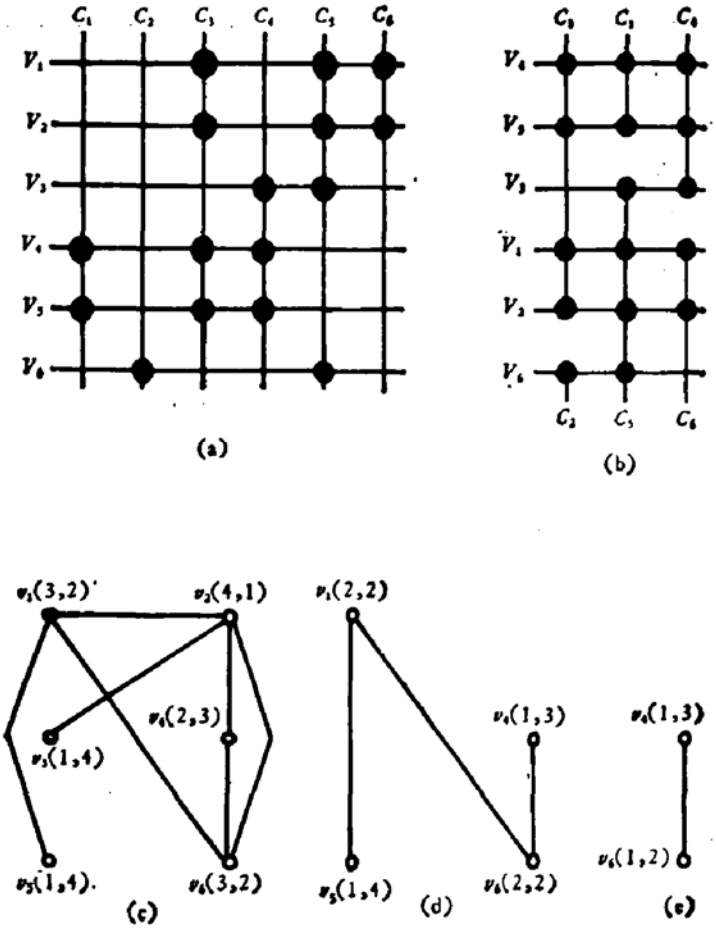

图 3 折迭示例

[注] $\left[\frac{a}{b}\right]$ 表示比 a,b 两数相除值小的最大整数

图 3 是一个列折迭演变过程的示例. 图 3(c) 是图 3(a) 所示阵列的折迭图，图中每个顶点边上括号中的数字表示它的折迭度和价格. 应用本算法，先后选中的折迭对是 (v_3,v_2)，(v_1,v_5) 和 (v_4,v_6). 图 3(d) 和图 3(e) 显示了每次选中折迭对后更新折迭图的过程，折迭后的阵列如图 3(b) 所示.

2. 折迭过程

在 FGPLA 的折迭程序中，采用与或与结构的阵列模式，折迭次序先后为：

1) 积项折迭　把积项取作符号矩阵的列，输入变量和输出变量取作行. 经过折迭，与阵列拆成两个，输入变量划分成两个集合，分属于这两个与阵列. 输出变量产生偏序关系.

2) 输入变量折迭　把输入变量取作符号矩阵的列，折迭过的积项作为行，对划分后的两个与阵列分别折迭，产生积项的偏序关系.

3) 输出变量折迭　根据已形成的偏序关系，把输出变量取作符号矩阵的列，对或阵列进行折迭.

3. 折迭参数法

我们用折迭后的阵列面积同原阵列面积之比称为折迭效率比. 在理想情况下，阵列的行和列都可以充分折迭，折迭效率最佳，达 25%. 在实例中，阵列中非零元的局部密集性往往会给折迭造成困难. 譬如有的积项价格大，含有较多的输入变量和输出变量，其折迭度自然也较小. 在积项折迭时，如果先选取这些积项，那么就会使划分在上、下两个与阵列集合中的输入变量个数相差较多以及输出变量的偏序关系较为复杂，从而容易造成循环约束，影响折迭效率. 又如某些输入变量的扇出量很大，则可折迭的积项将会很少. 特别地，如果某个输入变量的扇出量等于积项总数时，所有的积项都无法折迭.

为此，引进折迭参数 η 和 γ，把密集的行和列撇开，仅让分布较为均匀的阵列进行折迭，以改善折迭效率.

用 η 表示价格与总的输入变量个数比值的阈值，超过 η 值的积项认为是无法折迭的，不参加折迭. 把扇出量与积项总数的比值超过 γ 值的输入变量称为满变量，在积项中暂时把这些满变量除去，积项折迭后把它们同时加在两个与阵列中. 剔去这些积项和变量后，可望得到一个分布较为均匀的阵列，以得到较好的折迭效率.

在一实例中选取不同的 η 和 γ 值，得到折迭后的比值在图 4 中用曲线绘出.

$$\eta = \gamma = 100\%$$

相当于不用折迭参数划分原阵列的情况. 而当 η = 40—50%，γ = 30% 时，比值最小. 可见，η 和 γ 的引进可以使折迭效率得到改

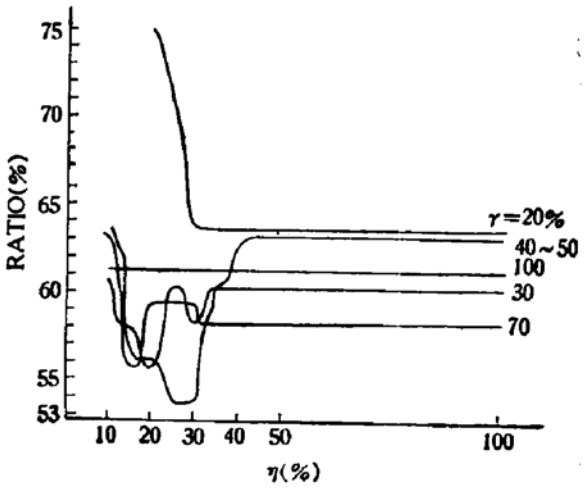

图 4　折迭效率曲线图

善．但是，η 和 γ 最佳值的选取强烈地依赖于各个实例本身．我们目前尚未从理论上得 η 和 γ 最佳取值的规律．根据经验在折迭程序中暂时约定 $\eta = 70\%$，$\gamma = 30\%$，并可由人工重置其值．

4. 交叉折迭法

积项折迭和输入变量折迭将对输出变量的折迭产生某些附加的约束条件，使折迭变得更加困难．为了在这些约束条件下提高输出变量的折迭效率，我们提出了交叉折迭的方法．

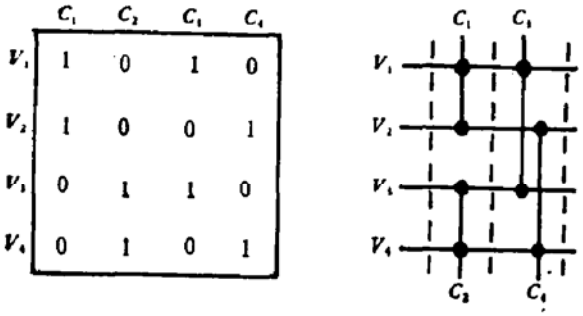

图 5 交叉折迭示意图

把输出变量看作待折迭的列．通常，如果有两个列 c_i 和 c_j 要折迭，那么 $R(c_i)$ 中的各行必须都位于 $R(c_j)$ 中各行的上方，如果做不到，c_i 和 c_j 就是不可折迭的．这是因为芯片中每一列的几何宽度只能容纳一个电路单元，其中包括一根纵向的连接线．如果增加一根连线，就要把这一列扩展为两列，失去了折迭的意义．但是，如果单元的宽度主要取决于单元中元件的几何宽度并大大于连线宽度，那么，将单元适量加宽，增加一条连线，就可以实现非零元不相交，布线重迭的交叉折迭(图 5)，从而改善折迭结果．

三、PLA 版图生成

折迭后的 PLA，可用各种工艺来实现，本文介绍了双极型版图的生成．首先，提出设计规则和适当的电路形式，由此设计出 PLA 的单元版图，然后根据折迭后的符号版图将单元版图的各种数据馈入版图生成程序，完成全部版图的描述．

1978 年，国内发展的大规模集成电路制版系统提供了描述电路版图的制版语言．为了提高版图生成程序的可靠性和灵活性，我们设计了一个介乎于 FORTRAN 程序和制版程序之间的版图生成设计文本．根据制版语言的要求，在该文本中描述了全部版图，由此可以非常容易地写出 FORTRAN 程序．由于这两种语言在结构上的类似之处，使该文本的编写提供了方便．例如，制版语言中用来描述版图子图形的积木块，其用法就与 FORTRAN 语言的子程序很类似．

1. 设计规则

版图生成的设计规则是根据线路要求、工艺条件以及制版语言的格式而制定的．

采用低功耗肖特基 TTL 电路和双层布线工艺，从而确定了各层光刻掩膜版和布线规则．

版图的最基本图形是矩形，要用四个数据来描述，称为图形的基本尺寸．版图中所有图形的基本尺寸，都是可以由符号版图以及表示工艺条件的版图参数确定的．我们定义了一系列变量来表示这些版图参数，并引进参数 λ 作为它们的单位．这样，根据不同的工艺

精度，适当调整这些变量的值，就可改变版图的尺寸，改变 λ 的值就可以实现版图的等比例缩放．在确定各种基本尺寸时，如果两个图形的间距需要同时满足若干个约束，则取其最大值．

2. PLA 的线路形式

根据 PLA 的与或功能，我们采用了简易的 TTL 与或非门作为基本电路（图 6）．同与或与形式的阵列相似，由两个与阵列、一个或阵列和两个电阻区组成．一系列多射极输入管完成与功能，各个发射极同与阵列中各个单元一一对应，悬空的表示空单元．一系列开集电极并联的倒相输出管完成“或非”功能，各个输出管同或阵列中各个单元一一对应，悬空的表示空单元．联结输入管发射极的水平线对应于输入线．联结输出管集电极的水平线对应于输出线，其中位于同一行中并行的输出线表示交叉折迭的情况．输入管集电极与输出管基极的垂直连线对应于积项线，其中上、下两个输入管的共基共集联结表示未折迭的积项线，此时下部的一个电阻悬空．

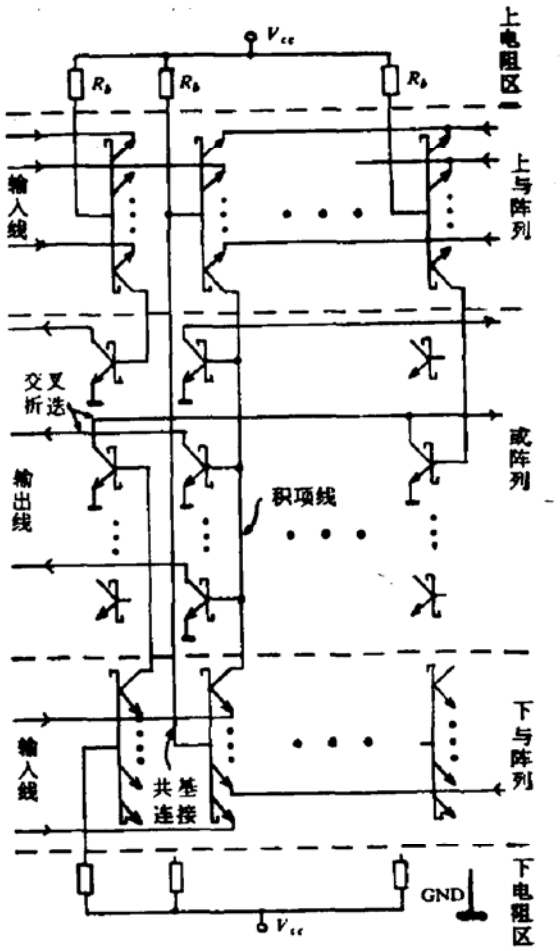

图 6　“与或与”阵列线路图

在“与或与”形式的阵列外面加上外围电路就可以形成完整的 PLA 线路图．外围电路包括输入缓冲器和输出驱动器，它们应能满足阵列电路的性能要求．

3. PLA 版图的划分和描述

由线路图、符号版图和工艺条件来设计 PLA 版图的过程，类似于数字系统的设计过程．即把整个版图先从大到小逐层划分成各级子图形，再从小到大描述各级子图形的版图，最后拼装成 PLA 的整图．

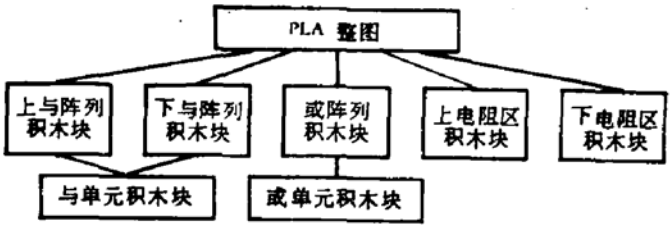

图 7　版图生成的层次结构

由图 6 的线路图，我们把版图划分成“与”阵列、“或”阵列和电阻区三大部分，再把“与”阵列和“或”阵列划分成若干种单元版图，这几层子图形都用制版语言的积木块来描述．两个与阵列、两个电阻区的版图虽然不完全相同，但描述格式是一致的，只要根据符号版图中不同的信息就可以得到不同的版图．最后，根据 PLA 的符号版图的数据来控制各部分子图形的调用和拼装（图

7).

四、FGPLA 的运行结果

表 1 列出了经过 FGPLA 运行的四个实例的有关数据. 经过折迭,阵列面积分别为折迭前的 55.81—60.22%. 表中的折迭参数 η 和 γ 都是多次运行后确定的最佳值. 这四个实例规模相仿,但由于各自阵列的密度分布情况不同,影响了 η 和 γ 的取值. 表内的运算时间是指对一确定的 η 和 γ 值折迭的运行时间, 逻辑综合和版图生成的运行时间不包括在内.

表 1 FGPLA 运行结果

	实例编号	1	2	3	4
折迭前	输入变量个数	16	16	16	16
	输出变量个数	8	8	8	8
	阵列行数	36	35	34	34
	积项线数	44	37	36	40
	有效密度(%)	19.4	17.9	18.7	19.6
折迭后	阵列行数	26	31	21	21
	积项线数	34	24	34	39
	折迭参数 γ(%)	30	30	40	100
	折迭参数 η(%)	40	100	70	100
	折迭效率比(%)	55.81	57.45	58.33	60.22
运算时间(秒)		24.6	22.6	25.6	21.9

参 考 文 献

[1] R. A. Wood, *IEEE Trans. on Computers*, C-28, 602 (1979).

[2] I. Suwa and W. J. Kubitz, Proc. of 18th Design Automation Conf., 398 (1981).

[3] J. F. Paillton,同上, 406 (1981).

[4] 黄均乃,赵文庆,复旦学报(自然科学版),即将发表.

[5] 洪先龙,钟龙保,唐璞山,黄均乃,大规模集成电路计算机辅助制版软件系统,国防工业出版社, (1981).

[6] G. D. Hachtel, A. R. Newton and A. L. Sangiovanni-Vincentelli, *IEEE Trans. on CAD of Integrated Circuits and Systems*, CAD-1, 63 (1982).

PLA's Folding and Automatic Generation of Mask Pattern

Zhao Wenqing and Tang Pushan

(Department of Electronics Engineering Fudan University)

Abstract

This paper presents an automatic design system for programmable logic array (PLA). It can implement PLA's design of multiple output combinational logic function with a total of thirty-two input and output variables. The design task consists of three procedures, i.e. logic simplification—synthesis, topological design—fold, and physical design—mask pattern generation. Two mapor procedures and some results are introduced in detail. First, a neww modified folding algorithm is described, and two methods, the folding factor method and the cross filding method aiming at improving the folding efficiency are presented. Second, the method of mask pattern generation for bipolar layout of PLA is presented. Finally, some results of the examples are lited.