

CMOS 门阵列的一种布线方法

张 钦 海 唐 璞 山

(复旦大学电子工程系)

1984年12月26日收到

本文提出了一种适合于单层铝布线 CMOS 门阵列的布线方法。对于不同的电路只要求改变铝布线设计就可完成。由于布线区域是固定的,通道中的走线密度不能超过系统预先给定的容量,同时,所有的信号连接端一般都须经多晶硅条方能引出,因而不允许存在两个或两个以上的不同信号网的连接端分配到同一条多晶硅上。为此,在本文中提出了一个信号连接点的线性规划算法和通道调整的子树连接算法,较好地解决了信号连接点在多晶硅条上的分配问题以及固定布线区域问题。

一、引 言

门阵列设计方法对于不同的电路品种只需设计一块铝布线模版,因而它具有设计周期短,设计费用低以及可靠性高等优点,在专用电路和集成电路系统设计中得到了广泛的应用^[1-3]。图1显示了硅栅 CMOS 门阵列的结构。其中内部单元由五对 CMOS 管组成,外围单元具有三态输出功能。单元之间存在水平通道和垂直通道。水平通道和垂直通道的相交点称为通道点。图2显示了单元间的相互连接情况。水平通道中预先做好的一组多晶硅条是供垂直连线使用的。

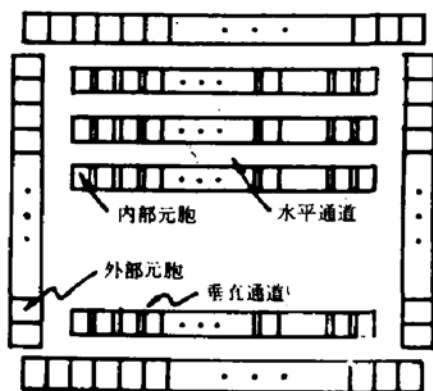


图1 CMOS 门阵列结构

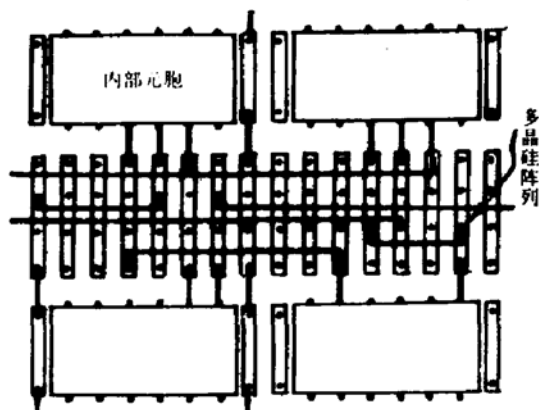


图2 单元间的相互连接

本文提出的布线设计方法是针对一层铝布线的模式,即水平连线是铝线,垂直连线由多晶硅条过渡。由于布线区域是固定的,同时所有的信号点一般都须经多晶硅条过渡才能引出。在本文提出的算法中,成功地解决了固定通道区域限制和信号连接点在多晶硅条上的分配问题。

本文提出的布线方法除基本单元外,还可以处理由若干个基本单元面积组成的宏单元,包括开放式(例如触发器单元)和封闭式(例如 RAM 和 ROM)宏单元。开放式宏单元内部相应区域允许走线,而封闭式宏单元则不允许走线。

二、连接树生成

门阵列的布线区域由水平通道和垂直通道构成。反映在水平通道中的信号网连接点包括单端引出和双端引出两种类型。由于双端引出中实际上已隐含了通过单元的内部连接,所以程序中允许有冗余点,即并非所有的连接点都必须连接到相应的网络上。

网的连接通常可表示为连接树的生成。目前已证明构造最优矩形斯坦纳树是属于 NP 完全问题^[4]。在多种的近似方法中,本文采用 Kruskal 的生成树算法^[5]以及相应的修正技术来完成信号网的连接。

设平面上 n 个连接点的集合 $V = \{v_1, \dots, v_n\}$ 。在 Kruskal 算法中所处理的仅是 n 个点的相互连接问题,不存在新的附加点(由矩形连接要求产生的斯坦纳点)和冗余点(生成树中双端引出集合中相应的单关联度点),边 e_{ij} 的选择只须考虑他们之间的距离。而本文所要处理的是存在附加点和冗余点的矩形连接问题。在算法中作如下修正:

1) 采用曼哈顿距离,并根据水平通道和垂直通道的容量大小,把水平线长和垂直线长分别赋以不同的权因子 IP_1 和 IP_2 。

2) 生成树的过程是子树不断合并的过程,这一过程是单向发展的,由于生成树的冗余点和相应的边可以被删除,所以在生成树的过程中,应优先考虑已经位于非空子树(该子树至少包含一个连接边)上的连接点的连接,即优先连接关联度大的连接点。采用因子 IP_3 ($IP_3 < 0$) 表示这种影响。

3) 在水平子通道中,过于密集的信号点的引出会导致多晶硅上重复占有态的上升,为使整个版图中信号引出端相对均匀分布,用因子 IP_4 来表示这种影响。

综合上面三方面的因素,边 e_{ij} 的权重 W_{ij} 定义为:

$$W_{ij} = IP_1 \cdot |x_i - x_j| + IP_2 \cdot |y_i - y_j| + IP_3 \cdot (d_i + d_j) + IP_4 \cdot (sc_i + sc_j)$$

式中 x_i, y_i, x_j, y_j 分别是 i 点和 j 点的坐标; d_i 和 d_j 是相应的点关联度; sc_i 和 sc_j 是 i 点和 j 点所在的子通道中的最大引出点数目。

在上述边权的定义下,逐次选择具有优先度高 (W_{ij} 值小) 的边生成连接网的树状结构 $G(V, E)$ 。由于在生成树 $G(V, E)$ 中,边集 E 可能存在不合理走线,点集 V 中有冗余点,因而还必须对不规则走线进行调整以及对冗余点进行删除。

三、信号连接点的线性规划

1. 问题的提出

信号连接点对多晶硅条的分配类似于计算机插件板上管脚对插针的分配^[6],文献[6]中是通过反时针向量的旋转和利用冗余度的概念来得到管脚对插针的分配。本文提出的算法则是利用线性规划理论对位于水平通道中的所有连接点,通过构造费用矩阵以及相

应线性分配问题^[7]的解来决定连接点在多晶硅上的一个分布。

在图 3(a) 所示的一个引线区域中, 设信号点的集合为 $S = \{s_i\}$, $i = 1, \dots, m$, m 是该区域中的信号点数目; 多晶硅条的集合为 $P = \{p_i\}$, $i = 1, \dots, n$, n 是该区域中多晶硅条的数目。现在的问题是如何选择一种信号点在多晶硅上的分布, 使得 S 在 P 上的分配在引线不相交的情况下达到最佳。具体地说就是多晶硅条上的重复占有率最小; 多晶硅条上的最大重复占有最小; 产生的引线总长度最小。

2. 问题的数学描述及解

设

$$\rho_{ij} = \begin{cases} 0 & \text{表示第 } i \text{ 个信号点不占据第 } j \text{ 根多晶硅条} \\ 1 & \text{表示第 } i \text{ 个信号点占据第 } j \text{ 根多晶硅条} \end{cases}$$

c_{ij} 表示第 i 个信号点引向第 j 根多晶硅条所产生的引线长度。分配优化的数学表述为:

$$\sum_{i=1}^m \sum_{j=1}^n \rho_{ij} \cdot c_{ij} \rightarrow \text{Min} \quad (1)$$

同时满足约束条件:

$$\sum_{j=1}^n \rho_{ij} = 1, \quad i = 1, 2, \dots, m \quad (2)$$

$$\text{Number } [b_j | b_j \geq 2] \rightarrow \text{Min}, \quad j = 1, \dots, n \quad (3)$$

$$\text{Max}[b_j] \rightarrow \text{Min}, \quad j = 1, \dots, n \quad (4)$$

式中 m 是信号点数目; n 是多晶硅条数目; $b_j = \sum_{i=1}^m \rho_{ij}$ 表示第 j 根多晶硅条上的重复占有度。

在门阵列单元电路设计中, 一般总能满足 $0 \leq m \leq 2n$, 我们取最大重复占有度为 2, 则约束条件(3)和(4)式可合并为:

$$\text{Number } [b_j | b_j = 2] \rightarrow \text{Min} \quad (5)$$

如果我们假设有 $2n - m$ 个虚拟信号点 $S' = \{s'_{m+1}, \dots, s'_{2n}\}$, n 个映照多晶硅条 $P' = \{p'_{n+1}, \dots, p'_{2n}\}$, c_{ij} 等价于第 i 个信号点引向第 j 根多晶硅条的费用函数, 那么我们就可以把上述问题归结为线性规划中的分配问题。采用 Munkres 算法就可获得相应分配问题的解。对图 3(a) 的分配结果如图 3(b) 所示。

3. 简并条件下的优化

在一个引线区域中, 定义上引出信号点集合为 $UPS \{s_i\}$, 相应的信号网集合为 $UPN \{n_i\}$, $i = 1, \dots, k_u$; 下引出信号点集合为 $DWS \{s_i\}$, 相应的信号网集合为 $DWN \{n_i\}$, $i = 1, \dots, k_d$ 。在实际中常常碰到的情况是 $UPN \cap DWN$ 不等于空集, 即表示 UPS 和 DWS 中存在具有相同信号网的连接点。由于相同信号网的不同连接点在同一多晶硅条上的重复占有可以蜕化为单重占有, 因而可以把本区域中相同信号网的不同连接

点分配到同一多晶硅条上,这样就可以改善引线区域中多晶硅条的重复占有状态。但是对属于同一信号网的第 i 个和第 j 个连接点,要求分配到同一多晶硅条 k 上,即 $\rho_{ik} = 1, \rho_{j,k+n} = 1$ 或 $\rho_{jk} = 1, \rho_{i,k+n} = 1$, 这对问题的解无疑是一种新的约束,并且在某种情况下成为不可解约束。在本文中我们用“简并”的方法基本上解决了这个问题。

对于信号点 $i \in UPS$, $j \in DWS$, 并且 i 和 j 属于同一信号网,构造一简并点 k_{ij} 代替 i 和 j 参加分配,相应的费用函数定义为:

$$c_{k_{ij}l} = (c_{il} + c_{jl}) \cdot IP, l = 1, \dots, n$$

其中 IP 是一加强定位因子。对所有满足简并条件的 i 和 j 连接点都进行类似的简并操作,然后对简并后的连接点集合进行线性分配。由于简并的存在,分配完毕后

$$(\rho_{il} = \rho_{jl} = \rho_{k_{ij}l}, l = 1, \dots, n)$$

可能导致最终结果中存在相交线和三态占有,这种情况称为不可解,为此应消除相应的简并态,重新进行分配,直到得到一个可行解,此时所得的分配就是考虑简并以后的优化解。

若定义多晶硅条的重复占有率为 λ , 则: $\lambda =$ 重复占有多晶硅条数目/总的多晶硅条数目。通过对大量例子的运算,考虑简并后, λ 值将大大降低。图 3(c) 显示了对图 3(a) 的分布采用简并方法所得的分配解。在图 3(b) 中,不考虑简并条件,重复占有率 $\lambda = 50\%$, 考虑简并后, $\lambda = 0$ 。

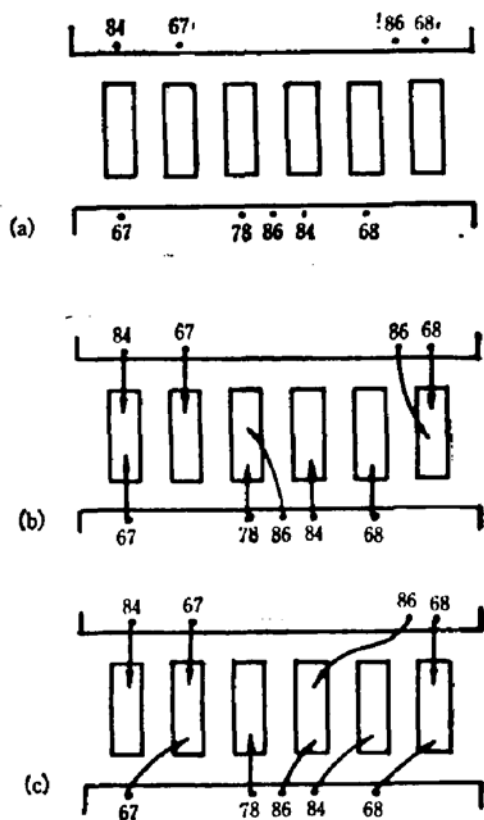


图3 (a) 信号点分布 (b) 无简并的分配解 ($\lambda=50$) (c) 存在简并的分配解 ($\lambda=0$)

四、通道分配和调整

1. 信号网的通道分配

信号网的通道分配是在不考虑通道容量的约束和信号网之间影响的条件下把初始形成的所有网的连线分配到相应的通道段中。

2. 子树的连接模型——通道点模型的建立

定义初始走线密度大于通道容量的区域为溢出区域。由于通道容量的限制,应对位于溢出区域的某些连接线段加以调整,使其经过其它通道来完成相应的连接功能。设当前调整线为 L_i , 去除 L_i 将导致原来的一棵树 T 分成若干个子树 $\{ST_1, ST_2, \dots, ST_l\}$, 传统的连接方法是以树枝 L_i 的两端点作为搜索的起点和终点, 在通道图中选择一条通路 P 连接这两点, 但这样做势必大大地增加连接树的长度, 并且在许多情况下, 使得这种搜索失败。考虑到整个网的电连通性, 本文提出了一个新的子树连接方法, 它是通过相应两

棵子树的连接来代替原来两点之间的连接。如图 4 所示。连接线 L_i 作为调整线所在的区域已发生溢出, L_i 不可能在原来路径上实现, 导致了原连接树 T 分裂成两棵子树 ST_1 和 ST_2 。通过采用子树连接方法我们可以找到一新的路径 L'_i 关联两子树 ST_1 和 ST_2 , 调整线 L_i 的作用被新的连接线 L'_i 取代, 去除 L_i 后, 原 L_i 所在通道相应的区域走线密度将降低, 从而实现了通道调整。

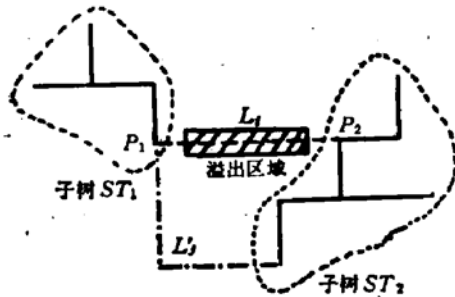


图 4 子树连接模型

(i) 子树的产生

设调整边为 L_i , 原来的连接树为 T , 在连线集合 $T - L_i$ 中, 若对于连接点 s_k 和 s_j , 存在着一通路 P_{kj} 关联于连接点 s_k 和 s_j , 则 s_k 和 s_j 属同一子树, 否则分属不同子树。按此规则形成子树集合 $ST = \{ST_1, ST_2, \dots\}$ 。

(ii) 子树向通道点的扩展

调整网的连接是通过水平通道和垂直通道来完成的。为了有效和迅速地连接子树, 把子树表示成为通道点的集合是十分重要的。定义 $HRD(j)$ 为连接点 s_j 的水平关联度 (连接点 s_j 所关联的水平线数目)。若 $HRD(j) = 0$, 则表示 s_j 点可同时沿 $-x$ 和 $+x$ 方向引到左邻和右邻的通道点上; 若 $HRD(j) = 1$, 则表示 s_j 点只能沿 $-x$ 和 $+x$ 方向引到相应的左邻或右邻的通道点上, 根据通道容量的限制及当前走线状态, 把子树向外扩展到相对应的通道点上。

(iii) 通道点模型的建立

子树扩展以后, 子树 ST_i 和 ST_j 的所有可能连接点都体现在相应的通道点上。设子树 ST_i 和 ST_j 对应的通道点集合为 CHP_i 和 CHP_j , 原来两子树 ST_i 和 ST_j 的连接就归结为在通道图中寻找一条路径 P_{ij} , 其始点 $SP \in CHP_i$, 终点 $EP \in CHP_j$ 。

(iv) 子树连接

由于子树 ST_i 和 ST_j 的连接可归结为在通道点集合 CHP_i 和 CHP_j 中选择一条通路, 本文根据李氏算法^[9]的基本思想, 提出了一个合理的子树连接方法, 其主要思想是: 在通道图中, 通道点通过水平通道段和垂直通道段相互键连。当子通道中当前的走线密度 $\geq$ 相应子通道的容量时, 则该子通道被封锁, 相邻两通道点之间的耦合断裂。如图 5 所示。因为子树可以表示成为通道点的集合, 若从属于起始子树的某一通道点出发, 如果该通道点存在键连, 则表明该通道点可以迷路发展到相应的邻近通道点上。和李氏算法的基本过程一样, 子树连接算法亦由布起始子树, 迷路发展以及回溯等过程组成。

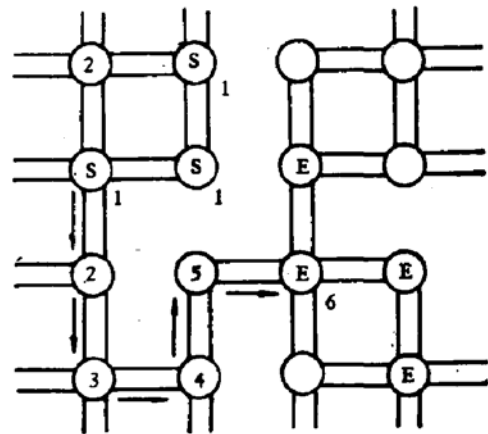


图 5 子树连接的实现

子树连接算法的特点是: (a) 可获得最优解。若两子树 ST_i 和 ST_j 之间存在通路

集合 $P, P = \{p_i^k\}$, 且 $|P| \neq 0$, 则该算法一定能找到一条关联子树 ST_i 和 ST_i 的最短通路; (b) 速度快、内存省。由于采用通道点结构, 可以使李氏算法中的内存需求量大, 搜索时间长等问题得以改善; (c) 调整效率高。在算法中, 由于采用子树连接方式, 即一个通道点集合向另外一个通道点集合探索的方法代替了过去常用的两单点之间的探索, 因而可以使可行路径的期望数增加, 调整的效率大大提高。

五、通道布线

对于一般的通道布线算法, 都要求在通道中主线所占的走线通路数最小。但在门阵列版图中, 通道中的通路数是预先给定的, 垂直约束关系将表现为不允许存在多晶硅条的重复占有。因此在本文所处理的通道布线中重点解决的问题不是使通道中的走线通路数最小, 而是在保证所有主线都能安排在通道中的前提下, 解决多晶硅条的重复占有和减少连线长度。

1. 水平通道预置线

对于消除多晶硅条的重复占有, 一个简单而有效的方法是把构成双重占有的某一端点不通过该多晶硅条而与其相应网直接相连。若把某些线网优先置入最低或最高通路, 与这些网相关联的重复占有就可以不通过多晶硅条而直接引出, 这就消除了相应多晶硅条上的重复占有。一个例子如图 6 所示。这些优先置入最高和最低走线通路的线称为水平预置线, 根据实际版图的情况, 通过预置的方法, 一般可以使多晶硅条的重复占有率下降 70% 以上, 对于多晶硅条的重复占有, 算法中还应用了信号点的直接引出, 间接引出以及交换引出等措施加以消除, 得到了预想的结果。

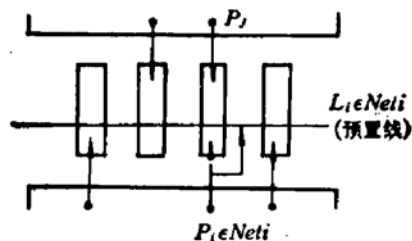


图 6 预置线消除重复占有

2. 水平通道主线分配——带权左边算法

在通道类布线算法中, 当主线不存在垂直约束时, 采用 Hashimoto 的“左边算法”^[3]就可获得最优解。

设力因子 f_H 为水平线 L_i 所关联的上连接点数目减去下连接点数目。为减少连线长度, 我们把水平通道分成上、下走线区域, 当 $f_H > 0$ 时, 应优先分配该线在上走线区, 否则为下走线区。显然, 当不存在线约束时(线约束是指预置线所产生的约束), 这种分区布线算法比单向布线算法具有更短的连线长度。

在连线比较密集的区域中, 水平预置线往往会成为上述带权左边算法的障碍线, 造成某些线在通道中无法安置。算法中是通过以下两个方法加以解决:

- 首先自右向左拆除已布线网, 将拆除线网在预置通道中试布;
- 解除预置线约束。

六、结 论

综上所述,同以往常用的一个网、一个网按次序串行布线方法相比,我们采用了一种块结构方式。在整个布线过程中,首先对所有的信号网在相同的条件下进行连接,然后对

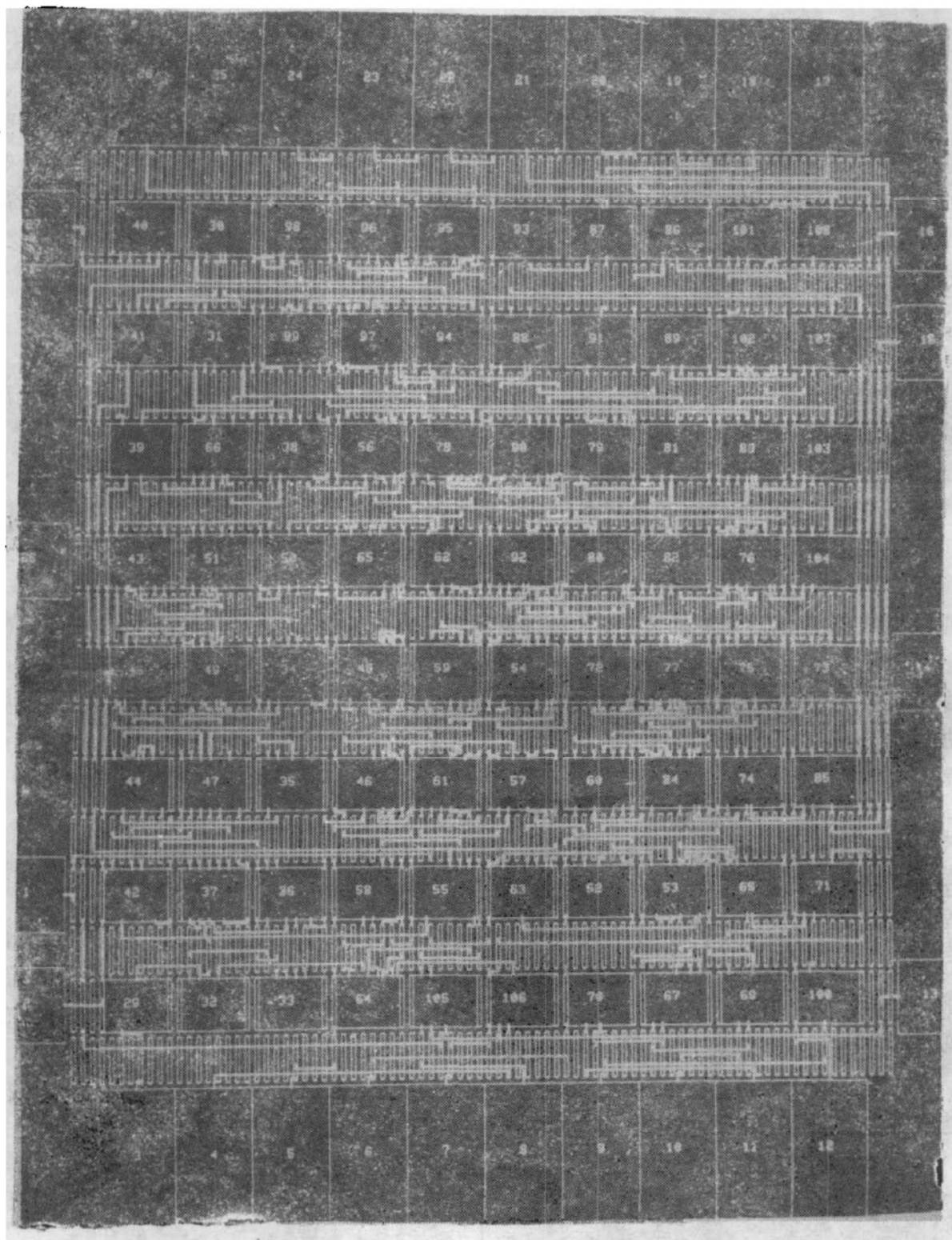


图7 心脏起搏器部分电路布线结果

信号点进行线性规划以及通道分配和调整,最后进行通道布线。这种把布线过程块结构化的层次设计过程,避免了串行布线过程中的线网排序,减少了先布网对后布网的影响,且因于通道调整总是在局部区域里进行,因而布线过程具有较好的整体性。

上述布线算法已编制程序 MSRP 并已在复旦大学 CMOS 门阵列版图设计系统中应用。表 1 给出了一个实际电路十个不同布局的布线结果。图 7 显示了一个心脏起搏器部分电路的布线结果。系统给定的水平通道容量为 6。

表 1 几个例子的布线结果

编号	总走线长度	平均密度 (水平通道)	平均密度 (垂直通道)	布不 通线数	布不 通点数	布通率(%)	运行时间(s)
1	57325	2.71	1.15	1	0	99.65	220.0
2	58750	2.79	1.20	0	1	100.0	217.0
3	51254	2.50	0.94	1	0	99.60	208.0
4	45619	2.24	0.76	1	0	99.55	199.0
5	50023	2.53	0.86	0	0	100.0	182.0
6	47932	2.36	0.90	2	0	99.13	191.0
7	59933	2.91	1.30	2	0	99.36	265.0
8	50636	2.60	0.96	5	0	97.95	227.0
9	49508	2.42	0.82	0	0	100.0	177.0
10	52694	2.75	0.92	3	0	98.82	260.0

作者对复旦大学 CAD 实验室的同志对我们的帮助表示感谢。

参 考 文 献

- [1] C. Tanaka, S. Murai, et al., Proc. 18th Des. Auto. Conf., (1981) pp. 812—819.
- [2] Tohru Adachi, Hitoshi Kitazawa, et al., Proc. 19th Des. Auto. Conf., (1982) pp. 785—790.
- [3] Hiroshi SHIRAISHI and Fumiyasu HIROSE, Proc. 17th Des. Auto. Conf., (1980) pp. 458—464.
- [4] M. R. Garry and D. S. Johnson, SIMA. J. Appl. Math., 1977, pp. 826—834.
- [5] J. B. Kruskal, Jr. Proc. Amer. Math. Soc., 7, 48 (1956).
- [6] M. A. Breuer, Design Automation of Digital System Theory and Techniques, Prentice-Hall, 1972, pp. 335.
- [7] J. Munkres J. SIAM, 5, 32 (1957).
- [8] Hashimoto, A. and J. Stevens Proc. 8th Des. Auto. Workshop, (1971), 214—224.
- [9] C. Y. Lee IRE Trans. On Electronics Computers, EC-10, 346—465 (1961).

A Router for CMOS Gate Array

Zhang Qin Hai and Tang Pushan

(Department of Electron Engineering, Fudan University)

Abstract

This paper deals with the single layer of aluminium routing technology for CMOS gate array. The main difficulties of routing are as follows:

1) Since the wiring area is fixed, local wire congestion is likely to occur, which may cause some of the nets to be unroutable;

2) Most of the terminals must be introduced with polysilicon, it's a restriction that two or more terminals from different nets shouldn't be assigned to one polysilicon.

In order to get a high quality routing, at first, a modified Kruskal's spanning tree algorithm used to find an initial globe routing for each net; secondly, according to the linear programming theory, the optimum distribution of all the terminals on the polysilicon array is obtained; then, a sub-tree connection method is selected continuously for the re-assignment of the unroutable nets until the routing density is less than or equal to the channel capacity; finally, the channel routing is accomplished by using the weighted left-edge algorithm and several heuristic methods is used to eliminate the multioccupation of polysilicon.