

具有单元自动生成的多元胞布图方法

蒋君伟 唐璞山

(复旦大学电子工程系, 上海)

1988年11月19日收到

本文提出一种新的多元胞自动布图方法。主要由四个部分构成, 块的生成、块内一维布局、单元生成、通道布线。其中第一部分采用了分析的方法完成各个块的生成, 目标为使连线最短和块之间连线和隔块连线最少。第二部分中引入了伪单元的概念以处理含有约束的一维布局问题, 解决了各个块之间的相互连线关系以及隔块连线。第三部分中的单元生成, 引入了类似硅编译的一些思想, 在硅编译与传统的自动布局方法之间的结合方面做一些有益的尝试。第四部分的通道布线是一个比较灵活的方法, 可以解决用户提出的各种工艺上的要求的布线, 提高了布图的物理性能。

整个过程用 C 语言编成程序并已在 PCS-68000 机上运行。

主题词: 自动设计, VLSI 设计软件, 自动布图系统, 多元胞系统, 单元自动生成

一、引言

多元胞设计模式发展已经相当成熟了, 目前已有不少实用的系统^[1,2]投入使用。在这种设计模式中, 各个单元的高度相同而宽度不同, 所有单元在芯片中排成行状, 因而最直接了当的布图方法便是先确定单元在各个行中分配, 然后再根据各个行间的关系完成一维布局。文献[3]中采用选取种子的方法, 然后再根据各个未安置单元与种子单元及已安置的单元之间的关联度依次将各个单元分配进各个行中。文献[4]中则是用等分接点法来完成单元的安置。一般则是采用二维布局方法^[5,6]直接完成单元布局。文献[5]采用相对布局来完成初始构造; 再通过一个“传输”模式完成布局。这是一种比较常用的方法。其缺点在于布局分为两个过程, 初始的相对布局的结果被改动较大, 从而影响了最终结果。文献[6]中用模拟退火的方法得到较好的结果, 但运行时间太长。文献[7]中则先将单元排成一列, 再根据实际芯片的大小将它截断以完成单元的安置, 方法直接了当, 且易于处理含有宏模块的布局。本文采用类似的思想, 并在划分和一维布局中提出了新的方法。先完成块的构造, 再进行一维布局和布线, 最后完成版图的设计。

二、系统设计思想

集成电路的布图设计是一个复杂的多目标的优化过程, 必须将它分为各个子过程分别处理, 各个子过程有各自的优化目标且应兼顾到总体的优化目标, 为了减少问题简化所

带来的影响，每个子过程的执行结果还必须要为下面的子过程的完成提供方便。根据这个原则，我们将多元胞布图过程分为四个子过程，并据此提出各个过程的优化方向。

(1) 块的生成 考虑布图面积随着连线总长变小而缩小这个因素，在块的生成中，尽量将单元之间关联度大的单元放到一个块中。因为块生成以后要进行一维布局及布线，因此块内单元之间关联度大，为本文中所提出的“生长法一维布局”提供方便。在布线时，垂直约束是影响布线面积的一个因素，而垂直约束则存在于块间的联线，因此，在块生成中，减少块间联线和隔块间联线可以为完成布线带来好处。本文采用点模型，将各个单元看成几何点，采用二次规划的方法来实现。

(2) 一维布局 虽然连线总长是一个影响面积的因素，但在多元胞模式中，布图面积由各个通道中的线轨 (track) 数目总和决定。尽管通道布线的优化目标是线轨数目最小，但在一维布图中就以此为目标，则可以使布局、布线两个过程更加有机地结合起来。而栅阵列 (gate matrix) 模式为实现这个目标给了很好的启示。对于块生成所形成的隔块连线，可以在一维布图中引入一个面积为零的伪单元参与布局。在布局过程中，各个块之间的相互作用则是通过块之间的连线来达到的。首先进行第一个块的布局，对它来讲，芯片上部压焊块 (pad) 的位置和两边的压焊块在布图过程中作为约束条件。而对于中

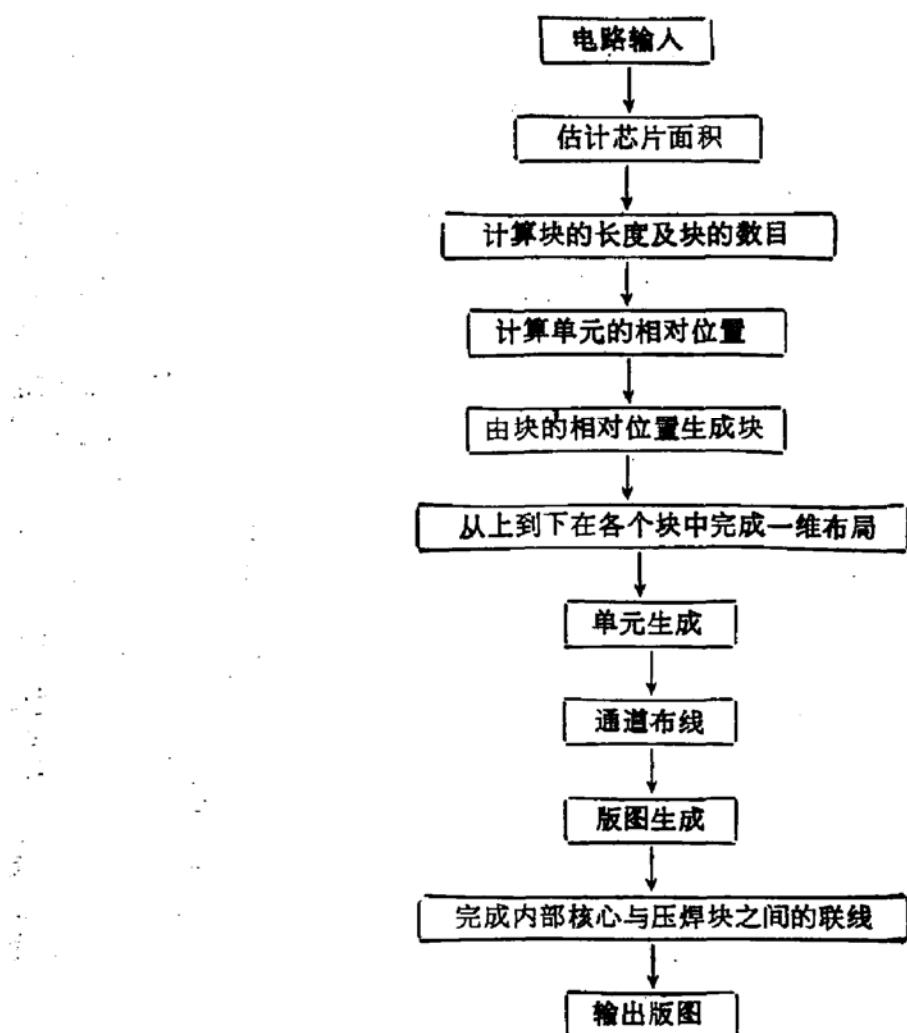


图1 布图过程的流程图

间的各个块来说，两边的压焊块和上边已布局单元与将布局块单元之间的联线引出头组成了这些块布局的三边约束。对于最下边的块来讲，上面已布局的块和下边的压焊块是它的上下约束，而两边的压焊块则是它的左右约束。对于隔块的线来讲，由于穿过块的那段线作为伪单元参加布局，因此与上面块的连接处则是这个伪单元的约束。

(3) 单元生成 它在整个过程起一个承上启下作用，使布局结果更有利于布线。因为在通道中进行布线时，必须根据通道两边的单元的引出脚排列顺序进行。因此，单元引出脚的分配对布线的优化结果有较大的影响。在以往的多元胞模式中，单元都是预先设计好的，在单元库中存放着各种单元版图，缺少灵活性。而在本方法中，只在单元库中存放单元内部的连接信息，在使用时，根据线路信息及与工艺独立的单元版图生成方法^[3]来给出单元版图，且单元的生成由布图的要求来完成，引出脚排列取决于布线的要求，这样就可以带来很多好处。第一，按照单元各引出脚所对应的线网在通道中走向来决定引出脚的分配，可以消除两个引出线在通道中相交错而必须占用二个线轨的情况。第二，根据布局以后通道中两边的引出脚情况，调整单元的引出脚分配，消除循环约束和减少垂直约束，使布线更为方便及易于完成。第三，在一定的范围内调整，可以使上下两边对应于同一线网的引出脚对准而不占水平线轨。由于布局采用点模型带来的误差，在此得到了消除。目前单元自动生成只限于各类比较简单的电路，以后将进一步扩充。

(4) 通道布线 采用混合图模型，处理对象灵活性很大，有对引线孔间隔要求，连线宽度，布线位置等各种要求^[2]。

三、算法介绍

下面按照流程的顺序依次介绍各个子过程的算法。

(1) 块生成的算法

采用分析型的算法，提出一个二次型的目标函数。以规划的角度阐明分划问题。提高了解的稳定性与全局性。

将单元在垂直方向分为若干块，实际上是将所有的单元按照一定的规律排成一列，且压焊块也作为单元处理，在一维排列中位置是固定不变的，按照各块的长度大致相同的原则进行截取，并要求在截取处的连线尽可能少。

单元排队的目标函数如下：

$$F = \sum_{j=1}^m d_j \left[\sum_{i \in N_j^c} (y_i - \bar{y}_j)^2 / n_j \right] / n_j \quad (1)$$

式中

$$\bar{y}_j = \sum_{i \in N_j^c} (y_i / n_j) = \frac{1}{n_j} \sum_{i \in N_j^c} y_i \quad (2)$$

$$d_j = d_{in} / d_{out} \quad (3)$$

$$d_{in} = 1 + \sum_{k \in M_j^a} (|N_j^c \cap N_k^c| / n_k) \quad (4)$$

$$d_{out} = \sum_{K \in M_i^c} (1 - |N_i^c \cap N_k^c| / n_k) \quad (5)$$

其中：

m 代表信号网总数， i 为单元编号， i 和 k 为线网编号。

n_i 第 i 个信号网中所有的单元数。

y_i 第 i 个单元的相对位置。

N_i^c 表示所有与线网 i 相连的单元集合，所以 $|N_i^c| = n_i$ 。

M_i^c 表示与 N_i^c 交集不为空的线网集合，即 $M_i^c = \{k | N_k^c \cap N_i^c \neq \emptyset\}$ 。

目标函数 F 的意义就在于使得各个线网中的所有单元都尽可能集中，使分划时块之间的连线减少。这个要求对某个信号网来说，就是要求该信号网所连的全部单元都在一个块中。显然，对于连接单元较多的大信号网来讲，要满足在一个块中则要占去很大部分面积，对于连接单元较少的小信号网来讲，满足的代价就小，因而在目标函数中加上 r 这个因子，以达到将大信号网拉开，小信号网集中的目的。实验表明， r 取 2 时分划的结果较好。目标函数中的 d_i 是用来表示线网的紧密度因子，因为对于某个线网来讲，其内部各个单元除了由该信号网相连以外，还可能有其它一些线网相连，因此对该信号网而言，其紧密度因子就大，这意味着如果该线网的单元紧密结合在一起会使另外的线网也同时满足要求或部分满足要求。

目标函数 F 不仅体现了分划的目的，还反映了连线总长最短这个布局目标。从(1)式中的 $\sum_{i \in N_i^c} [(y_i - \bar{y}_i)^2 / n_i]$ 可以看出它反映了线网内各个单元的相对位置的平均偏差，偏差最小，则连线最短，目标函数是正定二次式，且无约束条件（点模型），采用共轭方向法求解，使得分划的全局性很好，且与初始条件无关，稳定性很好。根据(1)式求得的各个单元的相对位置，然后将它们依次排列，依照块的大小进行截取，分块即可完成。

(2) 包含约束条件的一维布局算法

块生成以后，就可以依次在每一个块的内部进行布局，先布完的块的引出脚位置就可

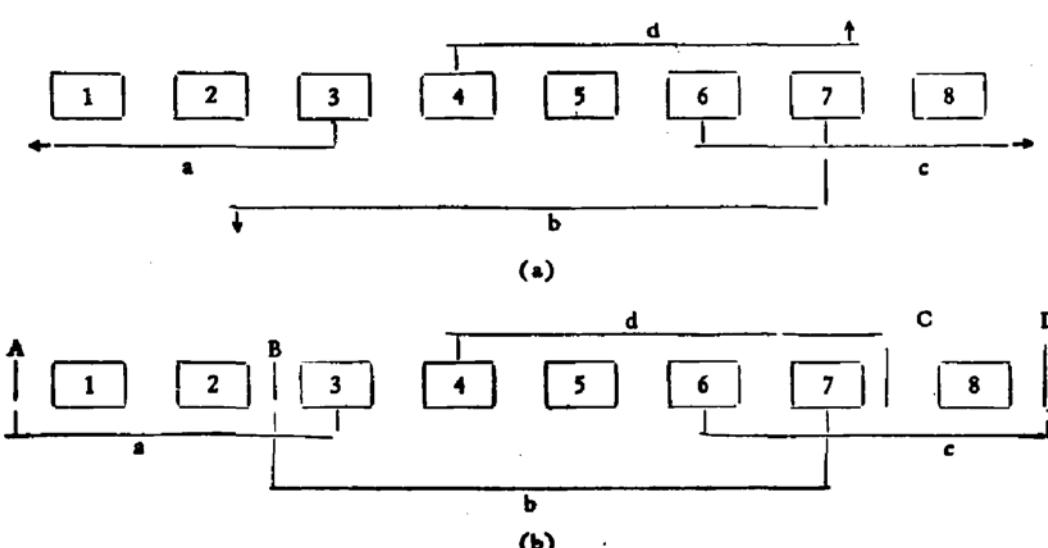


图 2 一维布局四面约束示意图

以作为未布局块在布局时的约束,具体可以通过一个伪单元实现,如图 2 所示。

在图 2(b) 中用 A, B, C, D 四个伪单元(物理位置固定而宽度为零的单元)来表示图 2(a) 中的约束条件。

首先是预处理过程,用来选择左右两边起始的单元。可以分为三种情形:a. 如在该块中左,右两边都有伪单元,则以它们为左右起始单元;b. 如该块中只有一边有伪单元,则在该块中选择与该伪单元有最长通路的单元作为另一边的起始单元;c. 如两边均无伪单元,在单元中选择两个具有最长通路的单元分别作为两边的起始单元。

算法过程如下:

```

main()      /* One dimensional placement */
{
    i = 0;
    Select starting cells Cr and Cl;
    track_No = MAX {out[Cr], out[Cl]};
    While (Cu != Ø) {
        if (there exist a pseudo cell Cp) {
            Put it at current position;
            end[Cp]--;
            if (out[Cp] > track_No)
                track_No = out[Cp];
        }
        else {
            /* Put cell on left side */
            Select a cell among Cu so that out[c] is minimum;
            if (there are more than one cell)
                Select a cell among them so that end[c]
                is minimum;
            /* Put cell on right side */
            Select a cell among Cu so that out[c] is minimum;
            if (there are more than one cell)
                Select a cell among them so that end[c]
                is minimum;
        }
        end[Cr]--;
        end[Cl]--;
        Cu = Cu - Cr - Cl;
        if (track_No < MAX {out[Cl], out[Cr]})
            track_No = MAX {out[Cl], out[Cr]};
    }
}

```

在算法中, $\text{end}[c]$ 表示安置单元 c 以后全部单元都已得到安置的那些线网的集合; $\text{out}[c]$ 表示安置单元 c 以后部分单元仍未安置的那些线网的集合(不包含所有单元均未安置的那些线网)。 C_u 表示未安置单元的集合。 C^l 和 C^r 分别表示当前被选中的左右安置单元。

本算法中,采用左右两边同时增长的方法,使得在布线时通道中的线网分配更均匀,且避免了将困难往后移的情形。

(3) 单元生成方法

对于 CMOS 的双端输出单元来讲,采用如图 3 所示的单元结构,其中的竖线表示多晶硅条,而两个水平的扩散区与竖线相交表示 PMOS 与 NMOS 管,且分别串联,对于这样一种结构,单元形成时,如果连接较复杂,则必然会需要隔离,也就会造成面积浪费.为此必须要合理安排管子的排列使隔离最少,用多重图 $G = (V, E)$

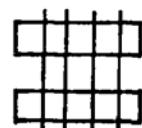


图 3 CMOS 单元结构示意图

来描述管子的连接.图中的一条边代表一个管子的源漏,点代表源漏之间的连接,如图 4 所示。

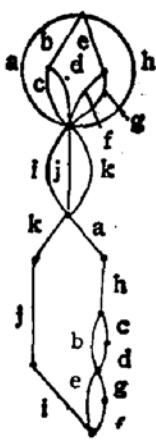
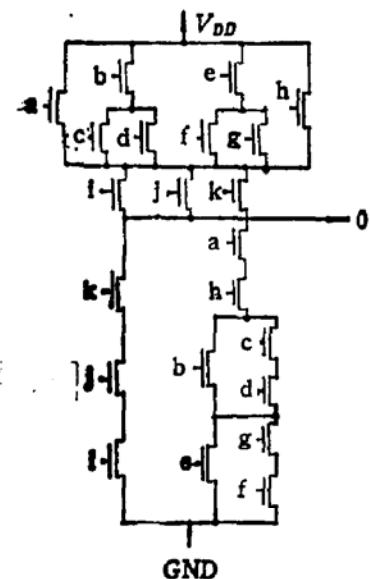


图 4 线路图及对应的多重图表示

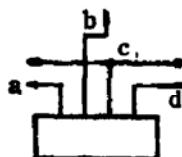
图 5 引出脚的外连类型

如果能在多重图中找出欧拉回路(Euler Loop)则可以按回路的秩序分配管子而不用隔离.对于一般情况来说,则需要找出它的覆盖集合 P , $P = \{P_1, P_2, \dots, P_k\}$, 其中, $P_i \cap P_j = \emptyset$, $P_1 \cup P_2 \cup \dots \cup P_k = E$. 显然, 单元所包含的互相隔离的子单元数为 K , 如果 $k = 1$, 则存在欧拉回路而不需要隔离.

定理: 在一个图中存在欧拉路径, 当且仅当图是连通的, 并且它只有两个(或没有)奇度数点.

引理: 对于偶数个并联的边, 其最优覆盖路径的起始点和终止点是同一个点. 在多重图中的每一条边就是每个管子的栅极, 在版图中就是多晶硅条, 也就是单元结构中的引出脚, 因此多重图中的每一条边就代表单元的一个引出脚. 在通道中单元(或子单元)的外连有四种类型, 如图 5 所示.

定义: 多重图中边的权重



$$d = \begin{cases} -1, & \text{Case a;} \\ 0, & \text{Case b and Case c;} \\ 1, & \text{Case d;} \end{cases}$$

$L, \sum_{j \in P_i} d_j < 0;$
 $M, \sum_{j \in P_i} d_j = 0;$
 $R, \sum_{j \in P_i} d_j > 0;$

定义: 子单元的指向 $D_i =$

由以上定理和引理, 我们就可以找出多重图的覆盖欧拉路径集^[10], 并且使得集合中路径数最少, 也就是子单元数最少。根据子单元的指向则可以合理分配它们在单元中的排列, 然后采用左边算法完成单元内部布线, 如有溢出, 则作为外连线布在通道中。算法如下:

```

/* Algorithm of Cell Generation */

main()
{
    /* Algorithm of finding the optimal set covering paths in a
       multi-graph */
    List all the even parallel edges and relative vertices in the
    multi-graph;
    Reduce the graph by removing these edges and vertices;
    While (There exists more than two odd vertices) {
        Choose one of them as starting point and go through the
        reduced graph until finding another odd vertex;
        'reduce the reduced graph by removing the path;
    }
    Finding the Euler path in the reduced graph by using the method
    in;
    Insert all the removing parallel edges back to the paths which
    have the common vertices with the parallel edges so as to the
    direction of the path and parallel edges are the same;
    /* Procedure of Path Assignment and routing */
    Sort the paths according to the value of  $\sum d_j$ ;
    Layout each path to the relative topologic cell structure from
    left to right;
    Adjust among the equivalent pins within the sub-cell;
    wiring within the cell by using the Left-edge Algorithm;
    if (the nets spill over the capacity of the track in the cell)
}

```

```

    route these nets in the channel as the external links;
}

```

图6给出了用本算法生成的对应于图4线路的一个单元拓扑图。

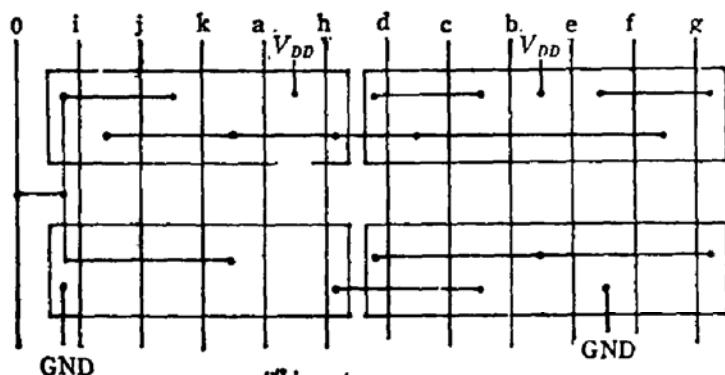


图6 单元拓扑图

(4) 通道布线算法

采用混合图模型算法^[9],在混合图中只对无向边标向,避免产生回路。拓扑布线后,可以按一定的设计规则展开成版图。反过来,根据工艺要求,例如引线孔大小,间距,连接宽度,间隔等。将其反映到混合图中,就可以在布图时就加以考虑,布线所处理的对象灵活性很大。

四、实例结果

整个设计过程已用C语言编制成程序,并在PCS, QU-68000机运行了若干例见表1。表中以 λ 为单位。

表1 实例结果

例子	1	2	3	4	5
单元数	40	23	56	78	225
等效门数	65	15	82	120	475
信号网数	35	19	51	73	238
块数	4	2	4	5	7
布线面积	360*177	168*75	424*207	496*264	1144*684
芯片核心面积	360*405	168*189	424*435	496*549	1144*1083
布图密度(3μ)(等效门/毫米 2)	198	210	198	196	170
CPU(s)	7.88	2.18	13.13	23.23	172.81

参 考 文 献

- [1] T. Kozawa, et, al., Proc. 11th, D. A. Conf., 26(1974).
- [2] K. Sato, et. at., Proc. 18th D. A. Conf., 828(1981).
- [3] 沙路,唐璞山,半导体学报, 5, 412(1984).
- [4] 程可行,庄文君,半导体学报 5, 422(1984).
- [5] Knut M. Just, Jurgen M. Kleinhans and Frank M. Johanners Proc. 23rd D. A. Conf., 308 (1986).
- [6] Loo K. Grover. Proc. 24th D. A. Conf., 56(1987).
- [7] E. M. Reingoldand K. J. Supowit, IEEE Tran. CAD-3, 3(1984).
- [8] Li Yingmeng Jiang Junwei and Tang Pushan, To be Published.
- [9] Jiang junwei and Tang Pushan, Proc. IEEE Asian Electronics Conf. 45(1987).
- [10] E. 米涅卡,李文强,赵关旗译,网络和图的最优化算法,中国铁路出版社, 226(1984).

A Polycell Layout Method with Automatic Cell Generation

Jiang Junwei and Tang Pushan

(Department of Electronics Engineering, Fudan University)

Abstract

A new polycell layout method is proposed. It is composed of four parts: row generation, one dimensional placement, cell generation and flexible channel routing. The authers use an analytical method to creat each row in the layout try to shorten the total length of wire and reduce the lines between rows and feedthroughs; introduce pseudo cells to deal with the restricted one dimensional placement to solve the connections between rows and feedthroughs; and adopt the idea of silicon compiler try to combine it with traditional automatic layout method. Because of the auotmatic generation of cells, the authers are able to overcome the shortcomings of the point model and get benefits for routing. The algorithm of the channel routing is based on the hybrid graph model, so it is easy to meet the needs of different IC techonologies demanded by users and make the layout more reasonable in physical property. The whole procedure is coded in C language and running on PCS-68000 computer.

KEY WORDS Design antomation, VLSI design software, automatic layout system, polycell system, automatic cell generation