

## LSIS-II 自动布图系统中的布局子系统

程可行 庄文君

(中国科学院半导体研究所)

1985年4月16日收到

本文介绍了我所研制的 LSIS-II 自动布图系统中的布局子系统。这是在[1]的基础之上,主要在以下的几个方面对布局模式和算法进行了发展。

(1) 布局子系统模块化,在布局过程的主要界面上可以再入,可以做人机交互,并可以由人工调度布局的进程;

(2) 在子系统中采用了宏单元插入的多元胞布局模式,构成可兼容多种布图模式的分级布局设计子系统,提高了系统的适应能力和设计容量;

(3) 本文的布局算法进一步提高了求解精度,兼顾了布局的拓扑目标和布线目标。

### 一、引 言

布局问题是大规模集成电路布图设计自动化的关键问题之一,布局的质量直接影响着整个布图设计的最终结果。多元胞模式是一种常用的布局模式,国外已有投入运行的多元胞自动布图系统<sup>[2]</sup>以及关于利用分级多元胞设计方式成功地设计了超大规模集成电路版图<sup>[3]</sup>。事实证明宏单元插入的多元胞布局模式是一种适应性和实用性较好的布局模式<sup>[4-6]</sup>。图1为这种布局模式的示意图。

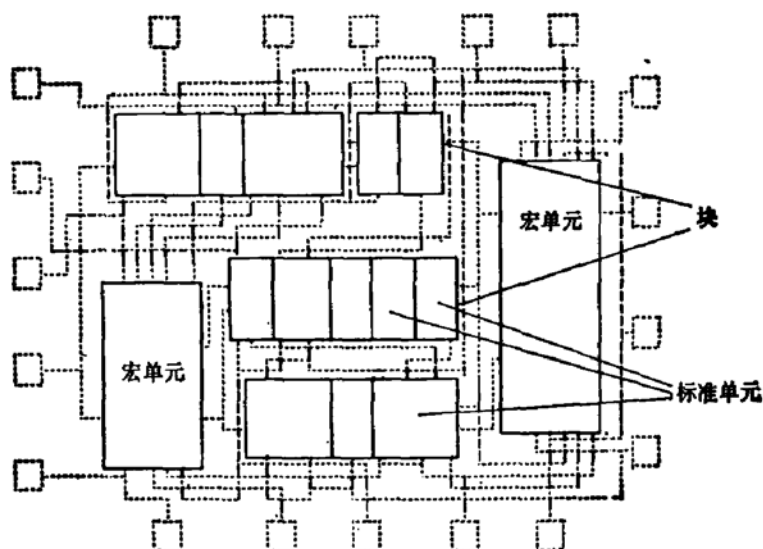


图 1

本文所指的标准单元,是高度等于一定值,长度自由的矩形,信号线从矩形的上、下边

引出。本文所指的宏单元,是高度、长度均为自由的矩形,信号线亦从矩形的上、下边引出。标准单元成行排列构成块,宏单元可跨越数个块。

之所以引入宏单元,是基于以下考虑:

1. 在较为复杂的逻辑电路中,往往存在一些内部连接关系复杂而对外连接关系相对简单的功能电路,如 RAM, ROM 等。由于电学性能上的要求,这类电路不宜按标准单元的高度来设计,引入宏单元,则可解决上述问题,提高了系统的适应能力。

2. 由于宏单元的高度、长度都没有一定限制,故可用多元胞模式或是 PLA, gatematrix 等手段自动设计宏单元,实现分级设计和多种设计模式兼容,提高了系统的设计容量。

3. 人工设计某些功能复杂的单元时,希望不要过分苛求单元高度的完全统一。宏单元的引入提高了系统的设计灵活性。

## 二、布局子系统的结构

布局子系统的结构如图 2 所示。

为了使问题简化,我们把二维平面上的布局转化为  $x, y$  两个方向上的一维布局问题。采用了把实际单元抽象成为点的拓扑构造以及复合目标函数多级迭代改善的组合算法。布局的各个子过程界面清晰,实现了模块化。

布局子系统有批处理和人机交互式处理两种工作模式。自动布局开始前,操作者可根据需要,如电路、工艺上的特殊要求预置单元;自动布局结束后,操作者可参考程序给出的布局评价参数修改布局。

由于系统实现了模块化和具有再入功能,因此操作者能够根据不同的设计对象和设

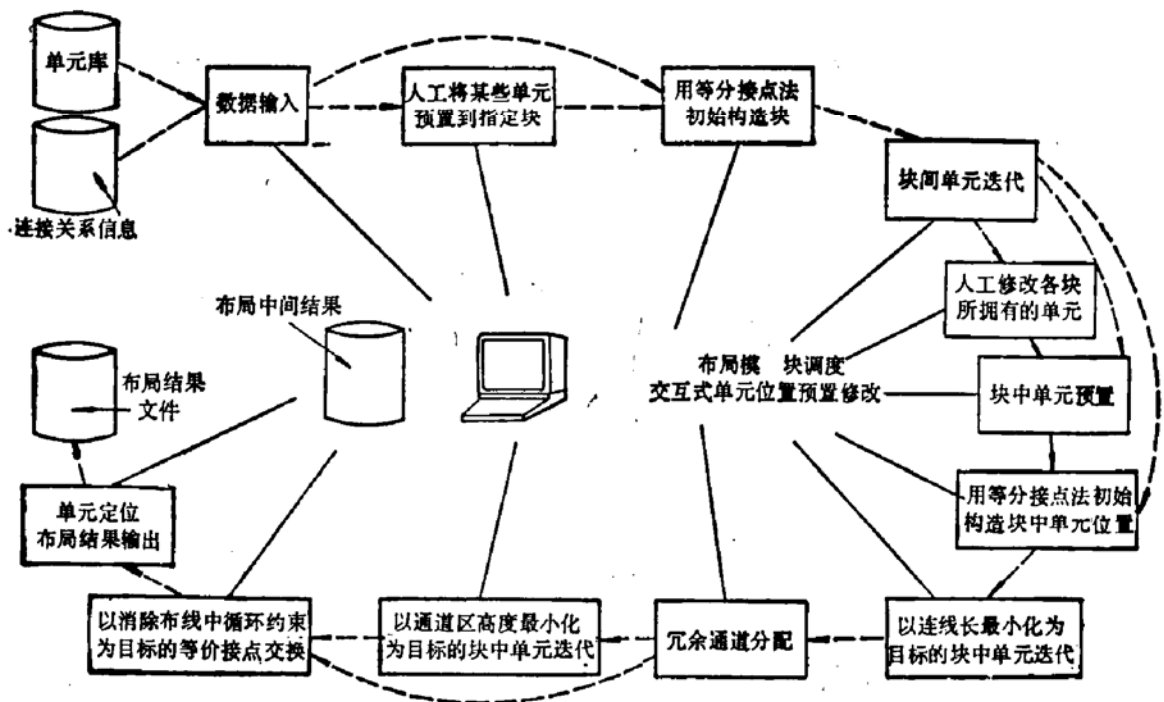


图 2 布局子系统结构示意图

计要求方便地使用各模块来处理数据,调度布局的进程,或令布局设计工作在任一子过程的界面上重新开始。这样,操作者享有充分的灵活性,对于某些性能要求不高的布图设计可不作系统中消耗机时多的处理,而对那些要求高,不吝惜机时的重要的布图设计,则可调用系统中具有更高求解精度的模块进行处理,甚至反复调用这些模块,或交替进行人工修改和程序自动处理,力求提高精度。若出于某种原因,譬如人工修改的失误,或模块的调度不合理,从而使布局结果恶化时,操作者可废除当前的布局;承认以前的结果,让布局工作从前面任一子过程的界面上再入,而不必再从头开始设计。布局子系统将有可能既充分利用程序自动处理的高效率,又发挥人对图形的高度识别能力和设计经验,得到较好的布局结果。

### 三、关于布局算法

本文仅论及算法思想与[1]不同之处。

1. 由于用等分接点法以纵向连线长度最小为目标进行块的初始构造时,把实际上具有一定几何尺寸的单元抽象为点,误差是难以避免的。而安置于同一块中的单元的选择函数的计算误差将是构造过程中产生误差的主要原因。这种误差还会在构造过程中积累,即安置不合理的单元有一种趋势,使与之连接关系紧密的单元也过早加以安置,扩大了安置的不合理。为此,我们把从上至下的“单边生长”改为从上下两个方向利用等分接点法交替进行构造的“双边生长”,减小了上述误差的积累,从而使初始构造的结果得到了进一步的优化。

2. 由于宏单元跨越数个块,破坏了多元胞模式下单元成行排列的有规则的布局,因而造成布局上的困难。考虑到实际电路的特点和宏单元的应用背景,一般的版图中只会有少数几个宏单元,所以我们对宏单元的位置和迭代加以限制。

在块的初始构造中,我们采取了以下措施使原算法可以兼容宏单元的安置。

令  $P_{j1}$  为包含第  $j$  个宏单元  $M_j$  全部上边引出接点的虚拟单元,  $P_{j2}$  为包含  $M_j$  全部下边引出接点的虚拟单元。显然,  $P_{j1}$  和  $P_{j2}$  的相对位置必须受  $M_j$  实际高度的约束。宏单元的安置选择函数  $f(M_j)$  定义为  $P_{j1}$  和  $P_{j2}$  的安置选择函数  $f(P_{j1})$  和  $f(P_{j2})$  的加权和。即

$$f(M_j) = k_1 f(P_{j1}) + k_2 f(P_{j2})$$

$$\begin{cases} k_1 = 1 \\ k_2 = \frac{h}{H_j}, \text{ 当上边生长时;} \end{cases}$$

$$\begin{cases} k_1 = \frac{h}{H_j} \\ k_2 = 1, \text{ 当下边生长时。} \end{cases}$$

式中  $k_1, k_2$  是权函数,  $h$  为标准单元高度,  $H_j$  为宏单元  $M_j$  的高度。

这样,宏单元就可以和标准单元一起,根据安置选择函数进行安置。

由于等分接点法,双边生长进行块初始构造的特点,宏单元  $M_j$  被安置后,当构造  $P_{j1}$

(或  $P_{j_2}$ ) 所在的块时,  $P_{j_1}$  (或  $P_{j_2}$ ) 才被看作是已安置单元。所以  $M_i$  是随着构造的进程分两次对其它单元的安置选择函数发生影响的。

3. 我们已做的模拟实验证明, [1]所述的块之间单元迭代过程所采用的复合目标函数(芯片横向长度最小, 纵向连线总长度最小)是十分有效的。但在迭代过程接近结束时, 主要反映连线长度变化的迭代对象选优函数与迭代过程的目标函数的对应性变差了。这是因为连线长度的缩短与芯片面积的减小客观上并不完全一致。对于一个迭代过程来说, 二者往往并非同时收敛。为使迭代复合目标函数的主项, 即芯片长度收敛到最佳值, 迭代过程的设计应能反映上述的规律。为此, 我们安排了另一个与原有迭代过程的目标函数相同, 但迭代对象选择函数不同的子过程。它完全不考虑连线长度的变化而只追求芯片长度的减小。在程序的控制下或人工的调度下交替运用这两个迭代过程, 使迭代过程继续向目标函数的主项收敛, 进一步提高了求解精度。图 3 所示的曲线表示了程序对同一布局问题, 在初始解不同的情况下, 迭代过程中连线长度与芯片长度的关系。可以看到, 收敛过程将是一个非单调过程。

4. 在  $y$  方向布局过程中, 分别位于单元上、下边上的等价接点(包括电学等价接点和逻辑等价接点)的分配是否得当会影响纵向布局的合理性。因此, 我们在块间单元迭代过程中, 每次迭代前, 先以纵向连线长度最短化为目标修正等价接点的分配。

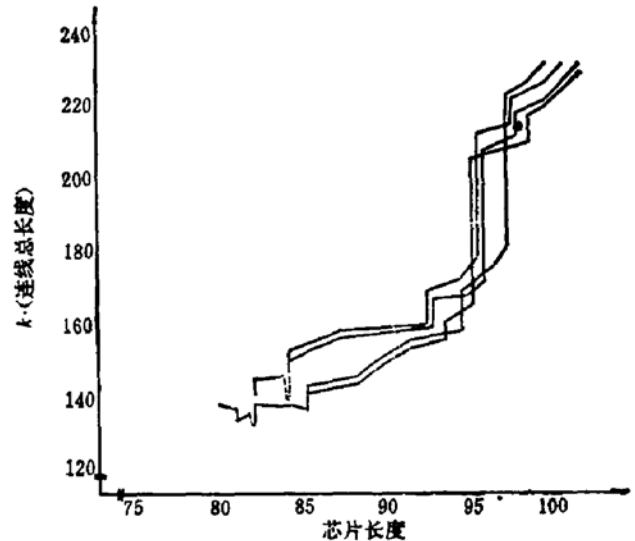


图 3

令线网  $i$  中接点自上至下的有序集合为  $(a_{i_1}, a_{i_2}, \dots, a_{i_n})$ ,  $U$  为单元上边引出接点的集合;  $D$  为单元下边引出接点的集合。

1) 如接点  $p \in \{t | t = a_{i_1} \Delta t \in U\}$ , 且接点  $q \in \{t | t = a_{j_1} \Delta t \in D\}$ , ( $j \neq i$ ), 且  $p$  与  $q$  是逻辑等价接点。

2) 如接点  $p \in \{t | t = a_{i_1} \Delta t \in U\}$ , 且接点  $q \in D$ , 且  $p$  与  $q$  是电学等价接点。

3) 如接点  $p \in \{t | t = a_{i_n} \Delta t \in D\}$ , 且接点  $q \in \{t | t = a_{j_n} \Delta t \in U\}$ , ( $j \neq i$ ), 且  $p$  与  $q$  是逻辑等价接点。

4) 如接点  $p \in \{t | t = a_{i_n} \Delta t \in D\}$ , 且接点  $q \in U$ , 且  $p$  与  $q$  是电学等价接点。则  $p$  与  $q$  做等价接点交换。

修正等价接点的分配不仅使最终布局的纵向连线缩短, 有利减小面积, 同时也避免了许多不必要的迭代。

5. 在块中单元迭代过程里, 也存在与 3 类似的问题。我们把块中单元迭代改为一个二级迭代过程。[1]中以  $x$  方向连线总长最短为目标的迭代作为第一级迭代, 用较小的代价先获得一较好的近似解; 另设置一个以布线通道区高度最小化为目标函数的迭代过程。

迭代逐个块进行。令块的各个单元, 包括作为过渡通道单元的块中冗余通道在块中

做单元插入<sup>[2]</sup>,每次插入后计算与该块邻接的两通道区的布线密度(density of channels),布线高密度列的广度(Span,以下简称通道区的广度.)及 $x$ 方向连线的总长度.若布线密度下降;或布线密度不变但通道区的广度下降;或前两者均不变但 $x$ 方向连线总长减小,则迭代成立.在计算布线密度时,考虑了由于宏单元的存在使通道区边界变得崎岖,对布线密度加以修正,使布线可较好地利用通道区中边界不规则部分.

因为在此迭代过程前已经过了一级迭代,所以在做单元插入时可仅在与插入单元当前位置邻近的区域内进行,以提高处理效率.

我们对若干个块分别以连线长度为单一目标函数和以布线密度,连线长度为二级目标函数进行了单元插入迭代,发现这两种不同的目标函数的迭代过程有相当好的一致性,但在迭代行将结束时,后者有可能使布线密度,即通道区高度进一步收敛.图4为对其中两个块所做的对照实验结果.图中虚线表示以连线长度为单一目标函数的迭代过程,实线表示以布线密度,连线长度为二级目标函数的迭代过程.

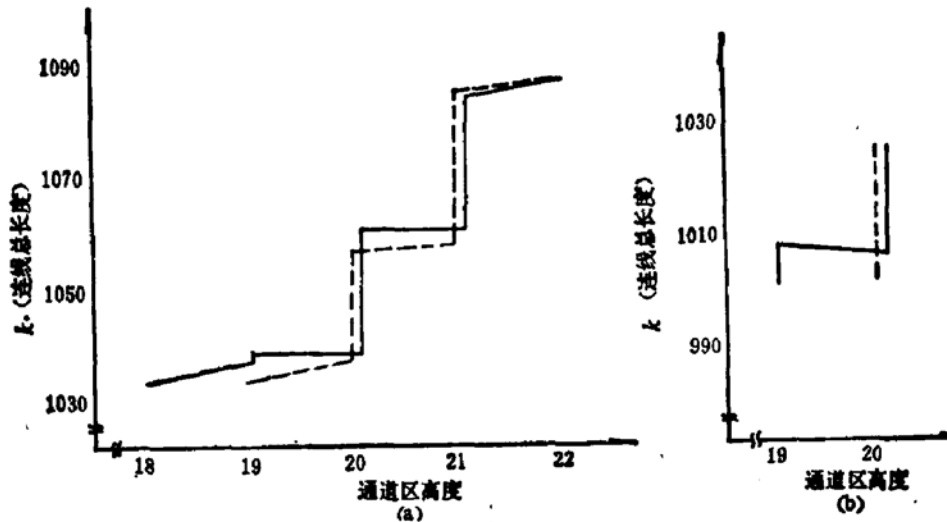


图 4

6. 在块中单元迭代改善时无论是邻接单元的成对交换或是单元的插入,其目的都是求得块中单元的合理的排列次序,使通道区的高度减小.然而通道区的布线密度不仅与单元的排列次序有关,还与块中各单元的具体位置有关.对于块中已完成排序的单元的位置进行调整,将是降低通道区高度的有力措施,有利于缩小芯片面积.<sup>[4]</sup>

本子系统具备上述功能.

7. 作为布线部分的前处理过程,在布局中要尽量考虑布线的要求是无可置疑的.我们所做的纵向冗余通道分配实际上就是在布局中已将全部穿越块的纵向连线布置完毕了.本文所述的以通道区高度最小化为目标函数的迭代企图降低通道区的布线密度和广度,则是着眼于提高通道区内布线的合理性.

影响布线的另一因素是布线垂直限制图中的循环约束<sup>[7]</sup>.我们分析了通道区布线中循环约束出现的一般规律和曲干布线(dogleg routing)时布线算法对循环约束的处理,为简化问题起见我们利用通道区中同一边上的等价接点(包括单元的逻辑等价接点,电学等价接点以及冗余通道)的交换,消除限制图中的循环约束.一般认为采用等价接点交换,

消除通道区中循环约束将是相当有效的。在等价接点交换中还兼顾了缩短连线长度。

#### 四、结 束 语

限于篇幅,本文着重介绍布局子系统的系统结构、功能组织和算法思想,未详细讨论具体算法。

布局部分作为 LSIS-II 多元胞自动布图系统中的一个子系统正在发展中。

图 5 为利用本子系统在 UNIVAC 1100 计算机上运行的一个实例。此例共有 150 个标准单元,2 个宏单元,148 条线网。布局所用 CPU 时间约为 3 分 48 秒。

本文的工作得到我所计算站,图书馆等有关部门的支持,在此表示感谢。

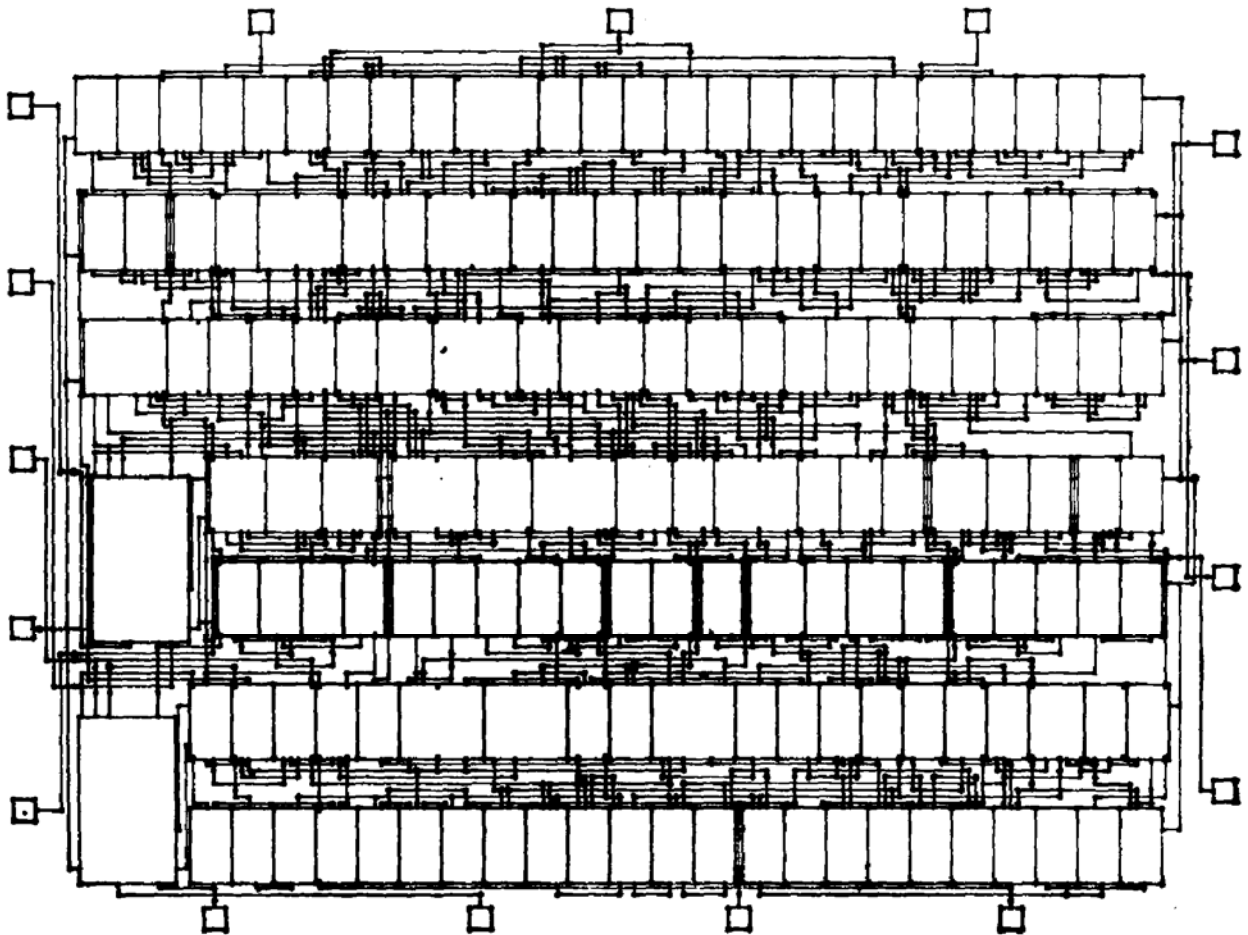


图 5

## 参 考 文 献

- [1] 程可行, 庄文君, 半导体学报 5, 422(1984).
- [2] G. Persky, D. N. Deutsch and D. G. Schweikert, 13th D. A. Conf., pp399—407 (1976).
- [3] S. Horiguchi, et al., IEEE Int. Solid-State Circuits Conf., 54(1982).
- [4] B. T. Preas and C. W. Gwyn, 15th D. A. Conf., pp. 206—212 (1978).
- [5] G. Persky et al., 18th D. A. Conf., pp. 22—29 (1981).
- [6] H. Beke and W. Sanser, 16th D. A. Conf., pp. 102—108 (1979).
- [7] 庄文君, 计算机学报 7, 217(1984).
- [8] 庄文君, 李玉兴, LSI/VLSI 布图设计自动化, 上海交通大学出版社(将出版).

**Placement Subsystem of the LSIS-II Layout Automated System**

Cheng Kexing and Zhuang Wenjun

(Institute of Semiconductors, Academia Sinica)

**Abstract**

The modularized placement subsystem for standard cell with macrocell is presented. The design can be in the batch or interactive way. In order to regulate the process of the placement, every module is reenterable. As macrocells are introduced, the layout of circuit with RAM, ROM, PLA and gate matrix will be feasible, and the hierarchical design will be realized. The multiple objective function and multilevel iterative method are adopted in improvement placement.