第12卷 第1期　　半 导 体 学 报　　Vol. 12, No. 1
1991年1月　　CHINESE JOURNAL OF SEMICONDUCTORS　　Jan., 1991

FPCS——一种适用于积木块方式的布局及平面规划系统

薄建国 俞明永 尹锦柏 庄文君
（中国科学院北京软件实验室，北京，100083）
洪先龙　　连永君
（清华大学计算机科学系，北京）　（北京集成电路设计中心）

1990年3月19日收到

本文简述了一种分级式的自下而上结群和自上而下分划定位相结合的全定制方式的积木块（building block）布局及平面规划（floorplanning）系统．本方法基于积木块的尺寸、形状、连接状况、引线位置以及芯片引线端等的要求逐级优化组合若干种积木块组，并且根据工艺条件进行了布线区面积估计，以便得到较好的布局结果．如果积木块的尺寸或其长宽比可以改变，则本系统可改变其尺寸及形状从而优化布局结果．由于采用了多种有效的实用方法，并把它们有机地统一在系统中，因此使布局能在基本满足用户要求的条件下，做到和布线结果基本匹配．实验结果表明，这种方法是令人满意的．

主题词　积木块方式布局，平面规划，全定制及专用集成电路设计

一、引　言

随着电子工业的飞速发展，VLSI 几乎已成了电子产品的关键性元件．用标准集成电路经过印制版连在一起来满足各种广泛用途的方法在很多情况下已不能适应用户的要求．近年来出现的半用户设计方法，仍然存在一些限制．据预测，用户设计的集成电路需求量将会有很大的增长．为了提高质量、缩短设计周期、降低成本，使用户专用设计电路在产量不大的情况下，仍然能够以适宜的价格来满足社会的需求，显然需要有一整套设计与生产的相应措施．本布局系统就是一个为此目的提供的商用计算机辅助设计系统中的一个重要环节．本系统是国家重点工程"熊猫"系统下的一个子系统，已于1989年12月完成原型．

二、特　点

（1）统一的使用方法和界面：FPCS 是"熊猫"系统的一个子系统，其使用方法和"熊猫"系统是一致的，用户无需做太多的学习就可应用自如．（2）人机交互的设计过程：用户可以使用一些成熟定形的模块（硬积木块）和一些尚未完全定型的模块（软积木块）来设计．只要用户给定软积木块的面积和长度比值的限制条件等信息，本系统可以提供这些

软积木块的较佳形状及尺寸等，以便用户做最终的选定.（3)分层的设计方式：在设计大系统时，用户可以先划分大系统为若干模块. 再对各模块用同样的方式划分设计；也可先设计各模块，再决定大系统. 每层的设计都可调用本系统. 本系统最多允许的积木块数为120，连线线网数为1100. 由于可采用分层的设计方式，更大的系统设计也可使用本系统进行设计.（4）灵活性与实用性：考虑到各种复杂的要求，FPCS 提供用户不同的算法及多种参数选择，以便用户为他的特定目标进行设计. 对于相当复杂的情况，从数据输入到布局完成，一般控制在两、三个小时之内. 输入数据的格式，也允许有不同的类型，例如：CIF，EDIF 等，扩大了系统的适用范围.

三、功能与特性

评价一种布局的好坏，是要在用户要求的面积和长宽比的范围之内，安排好各模块的位置，并在模块间相互连线完成之后，即使芯片面积最小，又使连线总长、最长的一些连线最短，同时保证那些影响线路电性能的连线长度符合要求. 不难理解，这些要求有时是矛盾的. 布局和布线实际上在设计过程中是应该统一考虑的. 但不幸的是，VLSI 的设计是如此复杂，以致于如果布局和布线同时考虑，设计时间之长、问题之复杂，实际上几乎是不可行的. 因此，实用化的 VLSI 布图设计（layout），都是把布局和布线分开来进行考虑的. 一个好的，较为合理的布局结果，不仅使布图结果能按用户要求予以实现（芯片尺寸，引出线端位置，线路的时间特性等），而且使布线在一开始就可在一个较为合理的基础上进行，各模块相对位置不会做较大的调动，布线也可顺序在各通道内依次进行，布线完成之后，也不要做由于布局明显的不合理因而推翻所有已完成的工作重新再布线. 这无疑减少了布线的复杂度，缩短了设计周期. 我们的布局系统，虽然主要是确定各模块的相对位置和方位，但丝毫也没有放松对那些布线要求的重视.

在 FPCS 中，采用了自下而上（bottom-up）和自上而下（top-down）相结合的方式. 这种方式的优点是把用户提供的信息逐级自下而上传至顶上，在顶上就可以借助于这些信息进行安置. 这使得布局从一开始就可在较好利用"资源"的基础上，尽量满足用户的要求[1].

所谓自下而上，这里指的是从积木块（block）开始，分析它们之间的连接度等因素，形成一些"团块"称为"群"(cluster). 基于计算复杂性的考虑，每个群均有一定的限制. 首先是一个群所含的积木块数目不能过大. 因此，结成的群内的子群就有一定数量的限制，然后把已结成的群，看成是一个个"伪"积木块，再重复上述过程，这样逐级"结群"，直至只形成一个群为止. 此时称为到达"根级"(root). 从积木块开始的那级称为"叶"级(leaf). 当自下而上的过程结束时，所要解决问题的资源也就基本上摸清了[2,3].

基于资源情况，根据用户要求，自根级开始，逐级向下做划分和安置，直至叶级为止. 这就是自上而下的模块布局，对每一级来说不仅考虑模块位置的安放，还要考虑模块间的布线区面积，找出一种最佳的安置方式[1].

上述的自下而上的结群和自上而下的布局，都是在假设积木块的大小是固定的，也就是硬积木块的方式下进行的. 为使系统兼容软积木块，在上述布局完成之后，做软模块的

形状优化：先在水平方向上，对那些影响芯片水平方向尺寸的软模块做水平方向的压缩．可以进行压缩的条件是：不增加芯片垂直方向尺寸，同时这样的软模块压缩后又满足用户给定的对这些软模块的形状限制．在垂直方向上，也做类似的事情，这样反复进行，直至不能压缩为止．最后就可给定软模块的最佳形状及接点方位[4,5]．

FPCS 还可评价已完成的布局．它可算出布局完成之后，芯片的水平及垂直方向的尺寸及芯片面积，水平及垂直方向上影响芯片尺寸的关键路径，模块间连线总长及最长线网的长度以及各布线通道的布线密度和通道容量的匹配情况（总体布线完成后）等等信息，这对布局作进一步改进及以后的布线是很重要的[5]．

四、主要算法思想

FPCS 提供了不同的结群算法，对于每种算法，同时还提供了若干参数供用户选择．它们是：两种匹配算法（matching）[6,7]，贪婪算法（greedy）[8,9]，随机算法（random）及人工结群（file-in）．提供人工结群算法，使用户可以人为地控制结群的方向，以使布局按指定方向进行．对一些特殊例子，人工进行干预，有时会得到较好的结果（表 1）．这一功能的提供，也为系统兼容其它结群算法，开了方便之门．对同一个待解决的问题，使用不同的结群算法，或虽然采用同样的结群算法，但施加不同的控制参数，一般说来，结群结果是不相同的．在积木块方式的布图（building block layout——BBL）中，不同的问题，所采用的积木块（block）的多少、尺寸、形状和它们之间的互连情况以及芯片引出端的位置要求，是很不相同的．例如在我们所采用的测试实例中，最少的积木块数为 2 个，最多为 111 个；同一个例子中，最大积木块的面积为最小积木块的 70 至 80 倍，而另外的例子中却相差不多；最多线网数为 1030，最少的才 2 条等等．因此，对不同类型和要求的例子，采用不同的结群算法，并施加不同的控制参数，就可以使结果尽可能向用户的设计要求靠拢．

表 1　对实例 iccad（8 积木块，22 线网）使用不同算法的布局结果

算　法	人工结群		贪　婪		匹　配　1		随　机	
目标形状个数	5	1	5	1	5	1	5	1
面　积	792825	813260	848859	848859	807570	807570	792288	825796

在 FPCS 中，不论是自下而上的结群或自上而下的布局，都考虑到这些群的形状和尺寸．即它们是按一定的“模版”存放的．所谓模版，其实是对一个长方形作一定的划分，划分出的每个小区域，称之为“房间”（room）．一个群由若干子群或积木块组成．这些子群或积木块如何安置到各个房间内，称为“房间分配”（room assignment）．如果一个模版中允许有很多房间，那么为完成各种可能的分配，就会耗费大量的机时．若每个群最多允许由 5（或 4）个积木块组成，那么 4（或 5）层可包含 125（或 256）个积木块．这里请注意，5 个积木块组成的群，已包含了 non-slicing 的结构[2,3]．做了这样的限制之后，选择房间最佳分配的穷举算法就会加快[2,3]，而且结群的层次也不会太大（本系统最大允

许模块数为 120). 我们知道,结群层次的加多,会使总的布局结果变坏[10,15]. 如果我们假定,对每种模版,均可做对称和 90 度角旋转变换,而且变换后产生的模版均不算作(或算作)新的模版种类(或类别),则各模版种类(或类别)和每个群所包含的积木块数(或子群数)之间的关系可见表 2.

表 2 模块种类和子群数的关系

子群数	1	2	3	4	5
模版种类数	1	1	2	6	19
模版类别数	1	2	6	22	92

模版类别数为: 模版种类经对称或旋转90度角变换后形成的模版算作新类别的模版

为使布局结果尽量和布线结果更好匹配, FPCS 还对各个群的面积作了粗略的估算. 以前的布线区面积粗略估算法的误差在 20% 左右[8], 如果在自下而上的结群中采用此算法,将会把自上而下的布局引向歧路,造成更坏的结果. 因此,有必要寻找一种既不过于复杂,又要比较准确的布线区面积估算法. 按传统方法,应当进行总体布线,构造 Steiner 树以寻找最短路径,只有知道连线的具体走向后,才能得到准确的布线区面积.这种方法,众所周知是很复杂的. 由于我们采用了有限的模版种类,所以可以事先依一定的几率,把一些走线信息算出来存在模版中,对于具体的连线情况,应用上述信息就可很快估出布线区所占用的面积. 对于一个具体的 BBL 问题,在叶子级,积木块的引线是已知的,可以利用这些信息;在其他各级,群内与群之间的连接状况也是已知的. 实践证明,这种做法,其自上而下和自下而上的面积估算偏差是令人满意的(表 3). 从表 3 中可知,模块数越多,结群级数也越多,其根级偏差也越大. 加大每个群的子群个数,对于相同模块数的例子,结群级数就会减少.

表 3 自下而上和自上而下面积估计的偏差

实例	iccad	ami33	modem	amd	ami49	4832	amd2	4930	ck2	txp
积木块数	8	33	5	17	49	20	47	27	16	111
线网数	22	121	36	288	476	647	399	359	109	1030
第一级偏差	0.060	0.008	0.056	0.040	0.004	0.003	0.046	0.059	0.036	0.046
第二级偏差		0.102		0.070	0.048	0.222	0.079	0.277	0.177	0.074
第三级偏差					0.362		0.509			—

FPCS 的自上而下的布局具有"向前看"(lookahead)的功能. 即如果自下而上的结群所提供的信息不能满足当前分划的要求时,可以再往下看一级、二级、……以便寻求其他的组合来满足当前的划分要求——在中途使自下而上和自上而下相匹配(meet in the middle). 不幸的是,这么做要花费大量的机时,这是因为寻求其它划分的算法是建立在穷举法的基础之上,对于较大的例子来讲,在实际处理中事实上是不可行的(时间长到几

乎不能容忍的程度）．为减少寻找的范围，用户可指定“修剪”值（prunning），即去掉某些搜索范围以加快处理速度．这种功能的优点及存在问题可见[1,8—14,16—19]．

为加快布局进程，改善布局结果，FPCS 中具有在结群处理中同时形成多个目标形状的功能（multiple target shapes）[2,3]．所谓目标形状，就是自下而上结群中所形成的各个群的形状尺寸．对每个群来说，其目标形状可以很多，我们从中选出若干个（目前系统中最多允许 5 个）供自上而下的布局进行选取，以满足其分划的要求．在产生多目标形状的算法中，剔除那些极易判断为不佳的组合，从而较大地缩短了处理时间．多目标形状的使用，使结果得到改善，并加快了处理速度（和 lookahead 比较）[1]．表 1 和 6 给出了一些实验结果．不难看到，采用 5 个目标形状后，比单个目标形状所得到的布局结果的面积有所减小，即面积利用率提高了．对于积木块数较多的大例子，改善更为明显，最大达到 36% 以上，而完成布局所花费的机时却增加不是很多．在[1]中给出的一些实验结果表明，采用多目标形状算法和“向前看”算法相比较，不仅提高了芯片面积利用率，减小了连线总长，而且缩短了处理时间．在这里，FPCS 比美国 U. C. Berkeley 的 BEAR 系统的优点，体现得更加明显．

表 4　自下而上结群，其根部浪费量计算值

实　例	iccad	intel	ami33	hp	modem	xerox	ami49	mar	test	txp
积木块数	8	62	33	11	5	10	49	20	4	111
线 网 数	22	570	121	83	36	203	476	656	29	1030
根级浪费量	0.301	0.534	0.256	0.301	0.037	0.148	0.414	0.232	0.093	0.508

表 5　使用 greedy 算法的芯片面积利用率

实　例	foo	modem	test	ami49	mar	iccad	txp	ami33	intel	rajiv
积木块数	2	5	4	49	20	8	111	33	62	17
线 网 数	3	36	29	476	656	22	1030	121	570	288
面积利用率（%）	100.0	90.3	87.1	83.3	80.7	78.4	76.8	74.6	68.4	46.1

我们知道，任何一个布局结果，一般讲都可由三部分面积所构成：模块占用的面积，模块间连线占用的面积和无用面积（dead area）．好的布局结果，应当使连线面积和无用面积尽量减少，这正是整个布局的目标函数之一．但对每一级，每个群或划分来说，其连线面积和无用面积却是局部量，因此可以不是最小的，但应该是较小的．判别这种无用面积的参考值，我们用“浪费量”（wasted）加以衡量，它是在自下而上的结群中形成的．它对于自上而下的安置中，选取怎样的目标形状（如果是多个目标形状的话），是有参考价值的[1]．为了使各个群的无用面积不会太大，我们对结群元素间的差异性作了限制．具

体讲，就是各元素的面积、线度比都不能超过某个界限．不难想象，在叶级的浪费量比根级小，而且越是浅层结群，浪费量也会越小．表 4 列出了一些实例在其根级，某个目标形状的浪费量．表 5 给出了某算法的面积利用率．影响面积利用率的主要因素是各模块的形状尺寸及其差异性以及模块间的连线情况．从众多例子中可见，最大面积利用率是 100.0%，最小为 46.03%，平均值为 77.91%．不难看出，浪费面积和面积利用率大体上是匹配的．

五、实 验 结 果

我们对大约 30 个实例作了各类算法，多种参数的实际布局处理，其中一小部分结果示于表 6．这些实例是从专门的 BBL 测试标准例子（benchmark）及工业界选来的．另外一些是我们构造的．表 6 中采用单个目标形状算法的布局结果是美国 U. C. Berkeley 的 BEAR 系统的实验结果．不难看到，对于积木块数较多的大例子，我们采用的多目标形状算法较大地改进了布局结果，最多可达到 36% 以上．对于 16 个例子的不同结群算法，其平均改善为 7.94%．由于 FPCS 采用多种有效的措施及其相互配合使用，结果是令人满意的，花费时间也不算长（VAX，SUN，HP 等机器），同时布局和布线的结果大

表 6 一些实例的布局结果

实 例	iccad								intel				ami33					
积木块数	8								62				33					
线 网 数	22								570				121					
引出端数	34								0				38					
算 法	贪婪		匹配(1)		随机		人工结群		贪婪		匹配(1)		贪婪		匹配(1)		随机	
NT	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1
T	131	127	75	73	116	80	56	8	2585	1513	2903	1689	1050	467	668	477	379	259
IMR(%)	0.00		0.00		4.06		2.51		1.43		4.47		13.24		9.17		2.03	

实 例	txp				ami49						amd2		amd		mar	
积木块数	111				49						47		17		20	
线 网 数	1030				476						399		288		656	
引出端数	239				24						48		0		139	
算 法	贪婪		匹配(1)		贪婪		匹配(1)		随机		贪婪		贪婪		贪婪	
NT	5	1	5	1	5	1	5	1	5	1	5	1	5	1	5	1
T	6011	3586	4627	3275	1758	600	557	411	610	434	3360	531	372	213	455	288
IMR(%)	27.97		36.25		7.41		10.39		6.20		13.99		19.9		14.84	

NT——目标形状个数 T——机时(cpu,秒) IMR——改善比率(%) IMR = (APS − APM)/APS(%)
APS——采用单个目标形状算法得到的布局面积 APM——使用多个目标形状算法得到的布局面积

致是匹配的，可见表 7，图 1 和 2．由于 FPCS 目前还没有处理 non-slicing 的能力，所以有些只当 non-slicing 结构才能组成最佳布局的例子，本布局系统将显得不够有力．

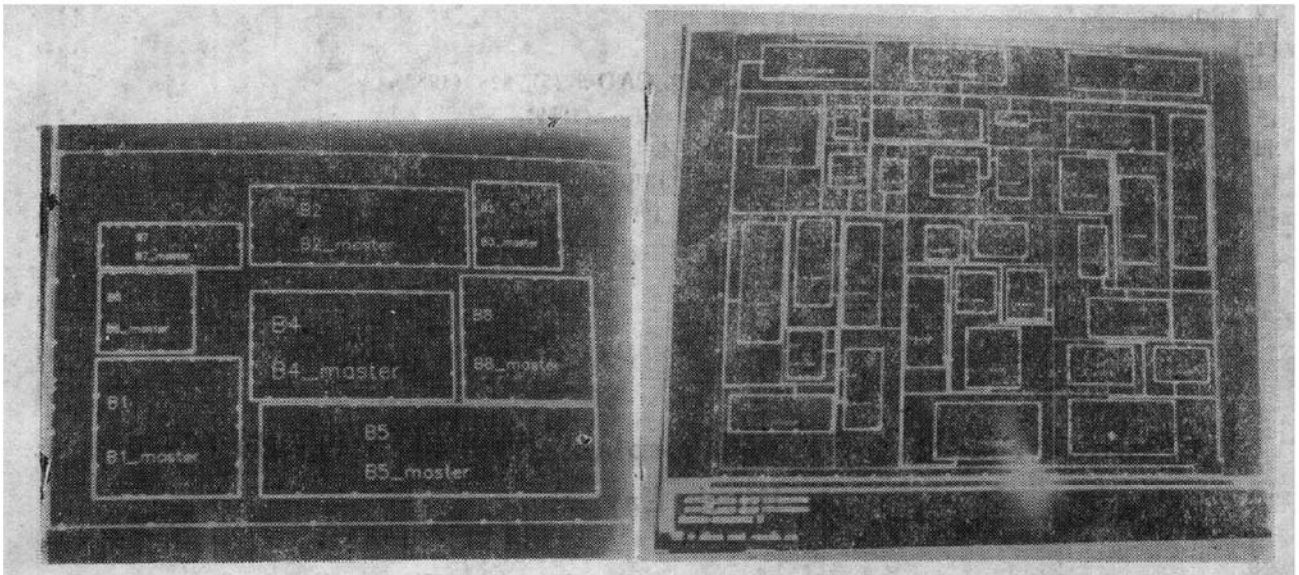

图 1　ICCAD 布局结果　　图 2　AMI33 布局结果

表 7　布局面积和布线后的芯片面积比较

实　例	iccad	ami33	apte	preas	xerox	ami49
积木块数	8	33	9	11	10	49
线网数	22	121	—	71	203	276
引出端数	34	38	—	33	2	24
Ap	923232	40978080	51888355	6990944	52734429	11329416
Ar	1013373	42398986	53006780	7097100	56483504	12293784
R(%)	8.895	3.351	2.109	1.495	6.637	7.844

Ap——布局面积；　Ar——布线后面积；　R＝(Ar—Ap)/Ar(%)

北京集成电路设计中心的同志们给予了极大的支持，刘伟平、张廷祥、单庆伟、王天泽、朱青、冯之雁、郑宁、孙国恩、戴得龙、邓学东、柯立刚等同志也给予很多帮助。参加本工作的还有吴强、李良斌等同志，对于他们的支持和帮助，谨致诚挚的谢意。

参　考　文　献

[1] 俞明永，薄建国，洪先龙，连永君，庄文君，半导体学报，11，609(1990).
[2] 薄建国，俞明永，尹锦柏，庄文君，洪先龙，"结群算法中多目标形状的产生方法"(待刊登).
[3] 薄建国，俞明永，尹锦柏，庄文君，洪先龙，连永君，"多目标形状结群及其在积木块布图设计中的应用"，全国第六届计算机辅助设计及图形学学术会议，1990.10.
[4] 尹锦柏，薄建国，俞明永，洪先龙，连永君，庄文君，全国第五届 IC-CAD 学术年会论文集，180—183(1989).
[5] 尹锦柏，"BBL 模式分级布局算法系统"，中国科学院半导体研究所，硕士论文.(1990.7)
[6] Davis Avis, Networks, 13, 475—493 (1983).
[7] R. E. Bukard *et al.*, Lecture Notes in Economics and mathematical Systems, springer-verlag, (1980).
[8] Bernhard Eschermann, "Hierarchical Placement for Macrocells with Simultaneous Routing Area Allocation", Master thesis, U. C. Berkeley, (1988).
[9] B. Eschermann *et al.*, Proc. Int. Conf. on CAD, 460—463, (1988).
[10] M. Y. Yu, X. L. Hong, Y. E. Lien, Z. Z. Ma, J. G. Bo, W. J. Zhuang, EDAC-90, 665—669, (1990).
[11] W. M. Dai and E. S. Kuh, "BEAR: A New Macrocell Layout System for Custom Chip Design", private

letter.
[12] W. M. Dai *et al.*, ICCAD'87, 34—37, (1987).
[13] W. M. Dai and E. S. Kuh, *IEEE Trans. on CAD*, CAD-6 (5), 828 (1987).
[14] W. M. Dai *et al.*, "BEAR Mannual", U. C. Berkeley, (1989).
[15] 俞明永、马佐成、庄文君，全国第五届 IC-CAD 学术年会论文集，201—204，(1989).
[16] Carl Sechen and Alberto Sangiovanni-Vincentell, IEEE Solid State Circuits, SC-20 (2), (1985).
[17] F. J. Kurdahi *et al.*, 23rd DAC'86, 467—473, (1986).
[18] X. Chen, 25th DAC'88, 54—59, (1988).
[19] G. Zimmermann, 25th DAC'88, 60—65, (1988).

FPCS: A Placement and Floorplanning System for Building Block Layout

Bo Jianguo*, Yu Mingyong*, Yin Jinbai*, Zhuang Wenjun*,
Hong Xianlong**, Lian Yongjun***

(*Beijing Software Lab., Academia Sinica, Beijing)
(**Department of Computer, Tsinghua University, Beijing)
(***Beijing IC Design Center)

Abstract

A hierarchical placement and floorplanning algorithm, combining the goal of blocks arrangement and orientation at top-down manner with the one at bottom-up clustering, are described. Based on the size, shapes, pin positions, connective relations of the blocks and pad positions of the chip, several optimizations of block groups with routing area estimated according to technology conditions at each level are created so that to reach the optimal placement. If the flexible blocks are included, the system can be optimized by resizing these blocks constrained to their areas or aspect ratio within their possible numbers. The placement and floorplanning result can be matched with the one of routing due to many effective methods in the system. Some results and comparisons with other methods are also given.

Key words Building block layout, Floorplanning, Custom and ASIC VLSI design