

# 基于位矩阵编码实现模拟集成电路 模块布局的遗传算法

张理洪<sup>1)</sup> 谢长生<sup>1)</sup> 张玉萍<sup>2)</sup> 裴先登<sup>1)</sup> KLEINE Ulrich<sup>3)</sup>

<sup>1)</sup>(华中科技大学外存储国家实验室 武汉 430074)

<sup>2)</sup>(上海师范大学机电学院 上海 200234)

<sup>3)</sup>(德国马格德堡大学 马格德堡 D-39016 德国)

**摘 要** 提出了一种新的实现模拟集成电路模块布局的遗传算法. 其位矩阵编码法提高了算法的搜索效率; 模块的滑行处理使绝对布局问题转变成相对布局问题, 极大地减小了搜索状态空间而不降低精度; 复制过程中个体间的相似性检查避免了算法的早熟收敛; 目标函数覆盖了模拟集成电路的特殊要求; 正交实验的方法用来研究算法参数, 其最优取值由另一个衍化遗传算法确定. 多种电路的测试结果表明, 该算法性能优于传统的模拟退火算法, 布局结果与手工布局相仿, 设计效率得到显著提高.

**关键词** 遗传算法; 布局问题; 参数优化; 模块; 模拟集成电路版图

**中图法分类号** TP391

## A Bit-Matrix Genetic Approach to Analog Module Placement

ZHANG Li-Hong<sup>1)</sup> XIE Chang-Sheng<sup>1)</sup> ZHANG Yu-Ping<sup>2)</sup> PEI Xian-Deng<sup>1)</sup> KLEINE Ulrich<sup>3)</sup>

<sup>1)</sup>(National Storage System Laboratory, Huazhong University of Science and Technology, Wuhan 430074)

<sup>2)</sup>(School of Mechanical & Information Engineering, Shanghai Teachers University, Shanghai 200234)

<sup>3)</sup>(Otto von Guericke University of Magdeburg, Magdeburg D-39016, Germany)

**Abstract** This paper presents a novel approach to analog module placement with genetic algorithm. The bit-matrix encoding is employed to improve search efficiency. It uses the principle of the divide-and-conquer technique, which allows the genetic algorithm to generate new configurations faster without degrading its search result. The idea of module slide is adopted to transform the absolute placement to the relative placement. It drastically decreases the configuration space without degrading search opportunities. Inspired by natural phenomena, similarity checks between the mating parents and between either parent and offspring are performed to improve the quality of evolution. They prevent the search from premature during the reproduction. The dedicated cost function covers the special requirements of analog integrated circuits. A fractional factorial experiment is conducted using an orthogonal array to study the algorithm parameters. A meta-GA is applied to determine the optimal parameter values. The fractional factorial experiment shrinks the configuration space so that the meta-GA can complete a search more efficiently. The Algorithm with optimized parameters is tested with several local benchmark circuits. The experimental results show this promising algorithm makes the better performance than the conventional simulated annealing approach with the satisfactory results comparable to manual place-

收稿日期:2001-11-22;修改稿收到日期:2003-05-07. 本课题得到撒克森—安亥州和西门子公司(2577A/0027B)资助. 张理洪,男,1971年生,博士,现做博士后研究,研究方向为优化排样以及大规模集成电路计算机辅助设计. 谢长生,男,1957年生,博士,教授,博士生导师,研究方向为计算机外设理论及技术、数据存储技术及多媒体网络存储技术. 张玉萍,女,1963年生,博士研究生,副教授,研究方向为CAD及优化技术. E-mail: ypingzhang@hotmail.com. 裴先登,男,1933年生,博士,教授,博士生导师,研究方向为计算机外设理论及技术、数据存储技术及多媒体网络存储技术. KLEINE Ulrich,男,1954年生,博士,教授,博士生导师,研究方向为模拟及数字电路设计、设计自动化.

ment. It considerably improves the design efficiency.

**Keywords** genetic algorithm; placement; parameter optimization; module; analog integrated circuit layout

## 1 引言

近年来,混合信号设计数量明显增多,大量新的集成电路(IC)迫切需要一个面向连续变化的外部世界的接口.这些设计中的数字电路部分可由发展成熟的开发工具进行合成、匹配或物理设计<sup>[1]</sup>,而模拟电路部分则只能由设计者手工进行.其计算机辅助设计工具仍处于发展的初级阶段,由于缺乏适合的开发工具,模拟集成电路部分已成为整个电路设计的瓶颈.

针对这一实际问题,我们开发出一个版图自动化设计工具——ALADIN<sup>[2]</sup>,它能将设计者的经验和知识融入合成处理中,最终产生高质量的模拟版图.本文的主题是关于其中的布局环节,由于模拟集成电路复杂的限制因素,如匹配要求等,将子电路中不同复杂程度的器件组成模块或宏单元是一种常用的方法.布局问题的核心任务是如何将模块合理放置,在特定的限制条件下使芯片面积和所有连线长度最小.

到目前已提出一些基于反复进步的启发式算法用以解决模块布局问题,如强迫导向法、最小切分法、被动电阻优化法<sup>[3]</sup>、模拟退火法(SA)<sup>[4~7]</sup>和遗传算法(GA)<sup>[8~10]</sup>.其中SA和GA是最晚得到应用的两种方法,SA已广泛应用于数字和模拟集成电路<sup>[6,7]</sup>领域,虽然效果较好,但耗时严重;GA的应用还主要局限于数字电路中.因此本文提出了一种新的基于位矩阵的遗传算法(BMGA),来解决模拟集成电路的模块布局问题,正交实验和衍化GA用于优化BMGA的参数.

## 2 BMGA 的实现方法

### 2.1 设计开发环境

ALADIN 提供了一个设计开发环境——设计助手(DA)<sup>[11]</sup>,DA被集成于一个商用设计软件中,其图形化用户界面可执行外部程序产生模拟版图,如模块生成器、布局器及布线器.图1给出了在DA中的设计流程,其中带有阴影的结构框为本文的实

际应用.在商用设计软件的原理图编辑器中输入一个仿真过的电路图后,多个模块被组建起来,电路图的分块过程是设计者经验和知识的直接反映.设计者在原理图编辑器中选择适当的元器件及对应的生成器,通过器件的匹配将网络表和参数由原理图传至模块生成器中.版图的寄生电容由一个电容敏感矩阵控制,模块选择和模块生成反复进行直到原理图中所有模块都被定义为止.模块布局时,从候选组中确定每一模块最适合的拓扑结构.版图生成后,寄生元件的提取操作将它们自动地标注到原理图中,然后整个电路可以在精确的寄生电容估计下仿真,通过改变电路的参数(参数优化)和/或重新定义电容敏感矩阵(应用优化),优化循环被执行.

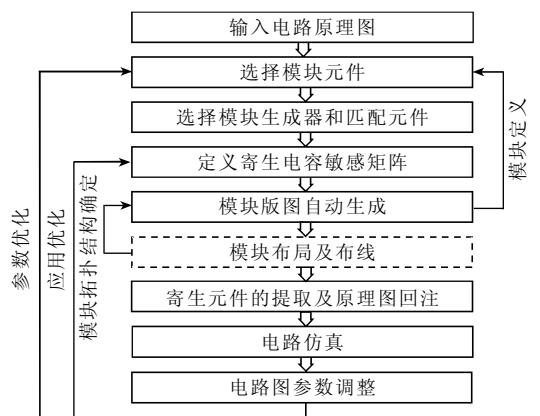


图1 DA中模拟集成电路版图的设计流程

### 2.2 染色体编码及遗传算子

传统的GA染色体编码一般是基于二进制串<sup>[8]</sup>,Chan提出一种灵活的位矩阵编码来处理数字电路的布局问题<sup>[9]</sup>,重新排列基因成为一个具有二维数组的染色体,这种分块-克服技术使GA能够更快地产生新的状态而不损失精度.我们提出了一种称为BMGA的遗传算法来解决模拟集成电路模块的布局问题,它基于位矩阵编码并融入了许多新的技术发展而成.每一个模块都表达成一个单行矩阵,其标准形式为

$$\{X_1, X_2, \dots, X_{N_x}, Y_1, Y_2, \dots, Y_{N_y}, O_1, O_2, O_3\} \quad (1)$$

式(1)中所有元素为1或0,二进制表示的三元组 $\{X_1, X_2, \dots, X_{N_x}\}$ , $\{Y_1, Y_2, \dots, Y_{N_y}\}$ 和 $\{O_1, O_2, O_3\}$

分别是模块的  $X, Y$  轴坐标和 8 种可能的方向(包括旋转和镜像),不同模块的表达式沿列排序,即模块数为  $M$  的布局状态可用  $M \times (N_x + N_y + 3)$  的矩阵表示.图 2 给出了一个模块数为 4 的染色体实例, $X$  和  $Y$  轴坐标用 3 位二进制数表示,即在图 2(a)中模块 1 位于(4:5)及方向为 5,模块 2 位于(3:4)及方向为 1 等等.坐标的表达位数应权衡搜索精度和效率确定,在程序中被作为变量,由设计者选择.

杂交(crossover)算子作用于被选出的双亲,两条随机的横纵切线如图 2,后代的左上和右下部分来自于一个双亲,左下和右上部分来自于另一个双亲.与 Chan 的算子定义<sup>[9]</sup>相比,杂交算子得到了简化,从双亲中只产生一个后代而非两个,就统计学角度,双亲四个部分的组合概率是相同的,因此简化不

会影响搜索精度.为提高搜索质量和效率,我们进行了许多尝试,对于杂交算子中的切线,提出了两种建议,其一是切点可以在任意位置,其二是切点只能在三元组的边界上,即双亲的  $X, Y$  和方向值只能传递给后代,无法通过杂交本身产生新的数值,实验结果表明第一种方法优于第二种.由于可能的逆转操作,在杂交操作之前须进行标准化处理,如图 2(c)中,双亲 B 经标准化处理与双亲 A 结构相同.变异(mutation)算子是将位矩阵中的任意一位取反,它提供了一种扩宽群体中基因取值范围的途径,保证了所有基因会在群体中出现.逆转(inversion)算子用于改变基因在染色体中的排序以削弱染色体中基因间的联系,其具体操作是随机地互换两行或列,逆转操作只改变染色体的形态而不改变其真实的布局结果.

	$X_2$	$X_1$	$X_0$	$Y_2$	$Y_1$	$Y_0$	$O_2$	$O_1$	$O_0$	( $X:Y:O$ )
$M_1$	1	0	0	1	0	1	1	0	1	(4:5:5)
$M_2$	0	1	1	1	0	0	0	0	1	(3:4:1)
$M_3$	0	1	0	0	0	0	1	1	1	(2:0:7)
$M_4$	1	0	1	0	1	0	1	1	0	(5:2:6)

(a) 双亲 A

	$Y_0$	$X_1$	$X_2$	$O_1$	$Y_1$	$X_0$	$O_2$	$Y_2$	$O_0$	( $X:Y:O$ )
$M_4$	0	1	1	1	0	0	1	0	1	(6:0:7)
$M_2$	0	0	1	0	0	1	1	0	1	(5:0:5)
$M_3$	0	1	0	1	0	1	1	1	0	(3:4:6)
$M_1$	1	1	0	0	1	1	1	0	0	(3:3:4)

(b) 双亲 B

	$X_2$	$X_1$	$X_0$	$Y_2$	$Y_1$	$Y_0$	$O_2$	$O_1$	$O_0$	( $X:Y:O$ )
$M_1$	0	1	1	0	1	1	1	0	0	(3:3:4)
$M_2$	1	0	1	0	0	0	1	0	1	(5:0:5)
$M_3$	0	1	1	1	0	0	1	1	0	(3:4:6)
$M_4$	1	1	0	0	0	0	1	1	1	(6:0:7)

(c) 标准化处理后的双亲 B

	$X_2$	$X_1$	$X_0$	$Y_2$	$Y_1$	$Y_0$	$O_2$	$O_1$	$O_0$	( $X:Y:O$ )
$M_1$	1	0	0	1	0	1	1	0	0	(4:5:4)
$M_2$	0	1	1	1	0	0	1	0	1	(3:4:5)
$M_3$	0	1	0	0	0	0	1	1	0	(2:0:6)
$M_4$	1	1	0	0	0	0	1	1	0	(6:0:6)

(d) 杂交后的子代

图 2 4 模块布局问题中的杂交算子

受自然规律的启发,双亲及子代之间的相似性检查帮助 BMGA 赢得了良好的质量.相似性检查由逐位比较得到,如果相同的位数在全体中的比率达到 85%,两种状态就称为过分相似.检查交配双亲之间的相似性,用来避免过分相似的双亲产生子代,类似地,检查子代和双亲的相似性,以保证在新生代中存在着群体多样性,如果子代和一个双亲过分相似并且它的目标值低于父代,该子代将取代父代,这样就保证了相似的子代和父代不会同时出现在新生代中,早熟收敛现象得到抑制.

### 2.3 模块滑行

布局问题可以分成两类:相对布局和绝对布局.理论上,无论是相对布局中的底-左布局<sup>[10]</sup>或分块结构布局<sup>[5]</sup>都无法相互取代,也不能包含所有可能的拓扑结构.虽然绝对布局在理论上能够包含所有可能的拓扑结构,但布局问题多状态空间特大的复杂度实际上阻碍了搜索的高效性.在 BMGA 中,模块滑行的处理将绝对布局问题转变成相对布局问题,当模块以绝对坐标定位后,滑行函数调整它们的

位置避免模块间的覆盖,使模块间的相对位置而不是绝对坐标变成了搜索的核心.模块滑行充分结合了绝对布局和相对布局二者的优点,可以包含所有可能的拓扑结构,并且保证了良好的搜索效率.

图 3 给出了模块滑行的一个示例,通常一个关键模块,如模块 A 被作为固定点,其它模块以它为参考点沿径向移动以避免覆盖,模块 B, C 和 D 顺序地按各自箭头方向移动,模块框中的数字为移动的顺序号.

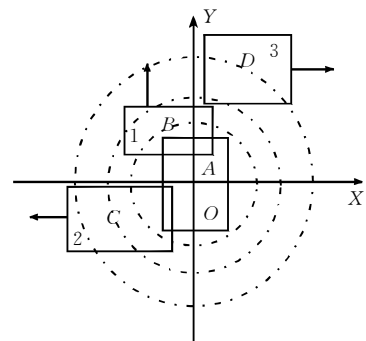


图 3 模块滑行示例

BMGA 伪代码算法流程如下所示.

### Algorithm BMGA.

( $M$ : the population size)

Begin

```

1 input module geometry and net-list;
2 initialize the first population randomly;
3 slide each member and evaluate the fitness;
   /* fitness 为对应目标函数值的倒数 */
4 while not (stopCriterion()) /* 判断算法的进化过程是否结束,条件是如果在一个预先指定的连续代数内搜索没有进展或指定的进化代数执行完毕 */
5   foreach (M * crossoverRate)
6     while (similarity1(two chosen parents)) /* 相似性检查 1,即检查两个被选出的双亲是否过分相似,若是则重复若干次双亲的挑选 */
7       choose the first parent based on the rank selection; /* 第一个双亲的选用用目标等级法,而非目标值法 */
8       choose the second parent randomly;
9     endwhile
10    do crossover operation to generate one offspring;
11    do mutation operation on the generated offspring;
12    do inversion operation on the generated offspring;
13    slide and evaluate the fitness of the generated offspring;
14    if (similarity2(offspring and each parent)) /* 相似性检查 2,即检查子代是否和双亲过分相似,并且目标函数值小于双亲 */
15      replace the similar parent with the offspring;
16    endif
17  endwhile
18  choose the best M members among the former members & new generated offsprings and set them as the new generation;
19 endwhile
20 output the best member;
End

```

## 2.4 目标函数(Cost Function)

目标函数是状态的优劣标准,它包含有需考虑的所有因素,并且控制着搜索过程,因此非常重要.我们的目标函数包含 4 个部分,如式(2)所示.

$$C = (\alpha_{\text{all\_area}} C_{\text{all\_area}} + \alpha_{\text{N\_area}} C_{\text{N\_area}} + \alpha_{\text{P\_area}} C_{\text{P\_area}}) + \alpha_{\text{nets}} C_{\text{nets}} + \alpha_{\text{size}} C_{\text{size}} + \alpha_{\text{overlap}} C_{\text{overlap}} \quad (2)$$

$\alpha_*$  是对应价值  $C_*$  的权重因子,它们用来根据不同的设计要求,权衡所有考虑因素的重要性.首先是面积价值,包含总面积和 NMOS, PMOS 面积,它用来使总面积最小并且 NMOS 和 PMOS 区域相对集中.其次是网络长度价值,不同的网络可指定为不同的优先级,一些敏感网络如差分输入信号,应该尽可能短以减小寄生电容和干扰,而其它非敏感网络,如电源和地线,可以适当长一些以避免网络阻塞.所以优先级因子被赋值给每一个网络,网络的敏感性

越高,它的优先级因子值越大.两个数学操作式(3)和式(4)可选择用来进行网络价值计算:

$$C_{\text{net}} = \sum_i p_i \times NL_i \quad (3)$$

$$C_{\text{net}} = \sum_i NL_i^{p_i} \quad (4)$$

$C_{\text{net}}$  是网络价值,  $p_i$  是第  $i$  个网络的优先级因子,  $NL_i$  是第  $i$  个网络的长度,对于比较敏感的网络,指数运算式(4)比线性运算式(3)施加了更大的惩罚量.网络长度的估算可使用 5 种方法,包括半周长法、质点法、完全图法、最小扩展树法和最小 Steiner 树法,电路设计者可选择合适的一种方法以权衡其特定的精度和效率要求.第三为尺寸价值,用来控制最后版图的形状,设计者可指定所希望的长宽比或具体的长度尺寸,实际版图尺寸与期望值相差越大,所导致的惩罚量越大.第四为模块覆盖惩罚值,由于滑行函数可被选择使用,这个惩罚价值只是在滑行函数被抑制时有效.价值函数的图形化输入窗口如图 4 所示.

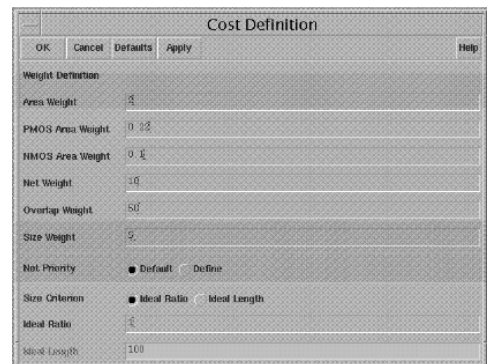


图 4 目标函数的图形化输入窗口

## 3 参数优化

遗传算子在进化过程中起着非常重要的作用,它们的参数,如群体大小、杂交率、变异率和逆转率以及敏感性和相关性等,将帮助我们确定在模拟集成电路模块布局问题中具体使用的参数大小.正交实验的方法用来研究参数的重要性及最优的取值范围,衍化 GA 用来确定最佳的参数取值,正交实验所确定的最优取值范围减小了衍化 GA 的搜索状态空间.

### 3.1 参数分析

正交实验是一种重要的健壮性设计技术<sup>[12]</sup>,它包含一组实验,所有实验数据集中起来用于分析不同参数对结果的影响.我们使用一个 Taguchi 正交表  $L_{27}(3^{13})$ ,群体大小、杂交率、变异率和逆转率作为因子.选择具有 13 列的正交表是为了对两两因子

之间的相关性进行全面的研 究,表头设计如表 1 所示,其中  $cr$  是杂交率, $mr$  是变异率, $ir$  是逆转率, $M$

是群体大小,\* 表示两个因子的交互作用,三列(4,7 和 11)空出作为误差估计.

表 1 正交表及表头设计

序号	实验 1	实验 2	实验 3	实验 4	实验 5	实验 6	实验 7	实验 8	实验 9	实验 10	实验 11	实验 12	实验 13
	$cr$	$mr$	$cr * mr$		$Ir$	$cr * ir$		$mr * ir$	$M$	$cr * M$		$mr * M$	$ir * M$
1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	2	2	2	2	2	2	2	2	2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
27	3	3	2	1	3	2	1	2	1	3	1	3	2

对于杂交率和逆转率,三水平取为(0.2,0.55,0.85),变异率取为(0.05,0.1,0.2),为了更大范围地覆盖群体大小,安排了两组实验,其一取(10,35,60),其二取(60,80,100).布局目标函数值和运行时间被作为优化标准,其中目标函数值比运行时间更重要.在执行了一组实验之后,收集这些实验数据并通过分析来确定不同参数的效果,实验分析基于统计方法,包括均值分析(ANOM)和方差分析(ANOVA).每一列有一个特定的自由度,某个因子均方和  $MS_j$  是对应的平方和  $S_j$  除以其自由度,如公式(5)

$$MS_j = S_j / f_j \quad (5)$$

其中  $f_j$  是该因子的自由度, $F$  比率是因子的均方和除以误差均方和,即公式(6)

$$F = \frac{(MS_j)}{(MS_{error})} \quad (6)$$

每一个因子的贡献百分比是该因子的平方和与所有平方和的比,即公式(7)

$$P_j = \frac{S_j}{S_T} \times 100\% \quad (7)$$

为了提高显著因子的分析精度,不是所有因子的组合都进行方差计算,由公式(7)估算出的不显著因子作为误差项,这样不考虑某个因子单独的贡献,而是使用它来调整对其它因子的贡献,被称为联营(pooling).最后计算出的每一个因子的  $F$  比率与在不同显著性水平  $p$  下的  $F$  分布临界值进行比较,来估计因子的显著性.

### 3.2 参数设计

因为正交表中因子的水平数有限,依靠正交表来确定参数取值是非常粗糙的,同时虽然某些两两因子之间的交互作用不太显著,在确定参数取值时最好还是考虑到它们之间的交互作用,因此我们采用了一个衍化 GA(meta-GA)来最后优化 BMGA 的参数.衍化 GA 本身就是一个遗传优化过程,它通过运行 GA 控制其参数以达到优化的目的.衍化 GA 中每一个体包含 3 个范围在 0~10 之间的整数,分别代表 BMGA 的杂交率、变异率和逆转率,根

据我们所进行的正交实验的结果,变异率的变化范围取为 0~0.1,步长为 0.01,杂交率和逆转率的变化范围取为 0.55~1,步长为 0.045,个体的目标函数值由一定参数配置下的 BMGA 运行得到,衍化 GA 根据进化过程中个体目标函数值搜索得到最优的参数配置.在衍化 GA 中,群体大小为 20,遗传代数为 100,杂交率为 1.杂交算子是等概率随机地从双亲中选择参数组合成子代的杂交率、变异率和逆转率.由于优化参数只有 3 个,几乎没有可能形成组间交互,因此没有使用逆转算子.在杂交之后,子代进行变异操作,变异算子是在子代的任一个参数中加上一个 0~10 之间的随机量,衍化 GA 的变异率取为 0.8,高的变异率和大的遗传代数使衍化 GA 不会成为 BMGA 参数优化的瓶颈.

衍化 GA 的伪代码算法流程如下.

#### Algorithm meta-GA()

( $M$ : the population size)

Begin

- 1 set the generation sum as 100, and  $M$  as 20;
- 2 initialize the first population randomly;
- 3 evaluate the *fitness*; /\* *fitness* 为对应目标函数值的倒数 \*/
- 4 while not (*stopCriterion*())  
/\* 判断算法的进化过程是否结束,条件是如果在一个预先指定的连续代数内搜索没有进展或指定的进化代数执行完毕 \*/
- 5 foreach ( $M$ )
- 6 make two random trials and select two parents from the population with the probability proportional to *fitness*;
- 7 perform crossover by selecting each parameter randomly from either parent with equal probability;
- 8 mutate offspring with the probability of 0.8 by selecting a parameter at random and adding to it a random number within the range of [0,10];
- 9 endfor
- 10 choose  $M$  individuals with the largest fitness among the combined set of parents and offspring;

```

11 endwhile
12 select the fittest set of parameters from the final
    population;
End

```

## 4 实验结果

程序用 C++ 语言编写,并运行于 Sun 工作站 (Ultra60) 的 Solaris-Unix 环境下. 目标函数权重因子的缺省取值为  $\alpha_{\text{all\_area}} = 2, \alpha_{\text{N\_area}} = 0.2, \alpha_{\text{P\_area}} = 0.2, \alpha_{\text{nets}} = 10, \alpha_{\text{size}} = 5, \alpha_{\text{overlap}} = 50$ . 由于目前国际工业界及学术界还不存在以模拟集成电路物理合成为目的的标准测试电路,考虑到集成运算放大器在模拟集成电路中具有重要的作用,我们首先用 3 个典型的运算放大器电路来评估以上算法并确定算法参数. 电路 1 是杆对杆运算放大器,包含 11 个模块和 16 个网络,电路 2 是标准的两级运算放大器,包含 6 个模块和 9 个网络,电路 3 是一个共模反馈运算放大器,包含 15 个模块和 17 个网络.

每一算法被运行 10 次取均值(mean)和标准差( $\sigma$ )作为估算数据,半周长法用所示,其中还包含模仿 Chan<sup>[9]</sup> 的 BMGA1 于网络长度的估算. 为了验证 GA 的高效性,也执行了一个 SA 的优化过程,结果如表 2(即无滑行函数和相似性检查),具有滑行函数、相似性检查和 8 位表示  $X(Y)$  的 BMGA2 以及具有滑行函数、相似性检查和 4~6 位表示  $X(Y)$  的 BMGA3. 实验结果表明 SA 比 BMGA 效果差,并且需要更多的运行时间. 由于搜索状态空间的减少, BMGA2 和 BMGA3 比 BMGA1 优秀许多,同时表达  $X(Y)$  的字节数也影响着结果,一般地,4~6 位工作得更好. 为了在进化过程中保持群体的多样性,个体选择采用目标等级法<sup>[10]</sup>,而不是目标值法(如滚动轮选择法),对于电路 1 的算法收敛曲线如图 5 所

示. 由于 BMGA 对应的目标值是每一代中的最优值,因此从整体上看,它们的曲线变化幅度比 SA 平缓,同时由于 BMGA2 和 BMGA3 中的染色体编码和滑行函数提高了搜索效率,使他们的搜索空间不必像 SA 或 BMGA1 一样覆盖很大的范围,而只集中在敏感区域进行高效搜索.

表 2 不同算法的比较

	SA	BMGA1	BMGA2	BMGA3
电路 1	目标值 <sub>mean</sub>	11799	13206	11693
	目标值 <sub><math>\sigma</math></sub>	176	655	110
	运行时间 <sub>mean</sub> (s)	846	1002	742
	运行时间 <sub><math>\sigma</math></sub> (s)	210	197	139
电路 2	目标值 <sub>mean</sub>	86653	112150	84735
	目标值 <sub><math>\sigma</math></sub>	3087	7883	3291
	运行时间 <sub>mean</sub> (s)	5068	3186	3196
	运行时间 <sub><math>\sigma</math></sub> (s)	783	620	376
电路 3	目标值 <sub>mean</sub>	215720	292240	202590
	目标值 <sub><math>\sigma</math></sub>	13554	8809	4818
	运行时间 <sub>mean</sub> (s)	1143	1321	1344
	运行时间 <sub><math>\sigma</math></sub> (s)	4	112	267

使用第 3 节介绍的参数优化方法我们对算法参数进行了研究,3 个电路的正交实验结果表明变异率和逆转率的交互作用对于运行时间十分显著,其它因子之间的交互作用不显著,群体大小在大于 60 时变得对目标函数值不再显著,杂交率和变异率对目标函数值十分显著,考虑与变异率的交互作用,逆转率最优取值为 0.85,因此 4 个参数的最后优化取值范围是群体大小为 60,变异率为 0~0.1,杂交率和逆转率均为 0.55~1. 基于正交实验所得到的参数最优取值范围,我们利用衍化 GA 的优化搜索过程来确定参数的最终取值,其实验数据结果如表 3 所示,最后参数取值如下:杂交率为 0.835、变异率为 0.023、逆转率为 0.805.

表 3 衍化 GA 参数优化的结果

	杂交率	变异率	逆转率	初始目标值	终止目标值
电路 1	0.82	0.02	0.73	12578	11325
电路 2	0.865	0.02	0.82	80247	77831
电路 3	0.82	0.03	0.865	210530	185276

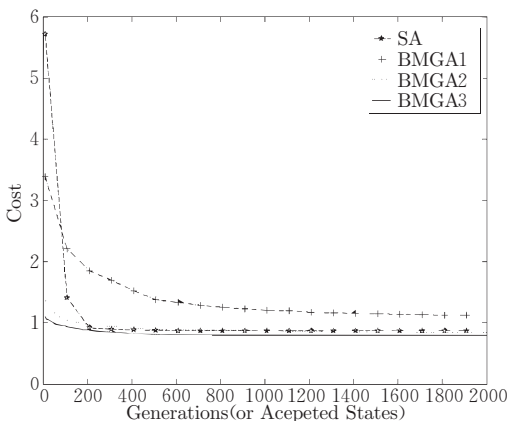


图 5 不同算法的收敛曲线

为了验证以上参数最优取值对一般电路的普遍适用性,我们使用这些参数又对另外 3 个电路的 SA 和 BMGA3 执行情况进行了对比实验. 电路 4 为一个高速 CMOS 比较器,包含 8 个模块和 14 个网络. 电路 5 为一个单级运算放大器,包含 7 个模块和 13 个网络. 电路 6 为一个低噪声运算放大器,包含 15 个模块和 21 个网络. 6 个电路的实验结果直方图如图 6 所示,图 6(a)为各电路运行 SA 和 BMGA3 所得目标函数值的比率,其中所计算的目标函数值已

去除了固定模块面积的影响,图 6(b)为各电路运行 SA 和 BMGA3 所花费时间的比率,从中可以看出

对于所有电路 BMGA3 的运行结果明显优于 SA.

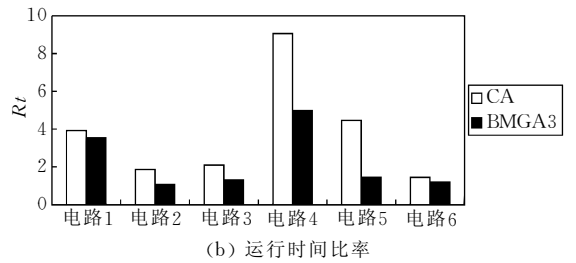
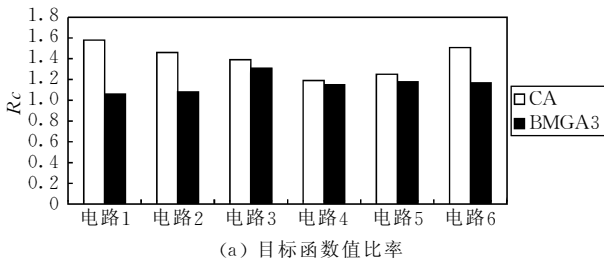
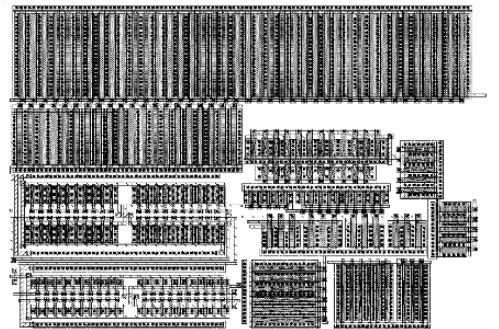
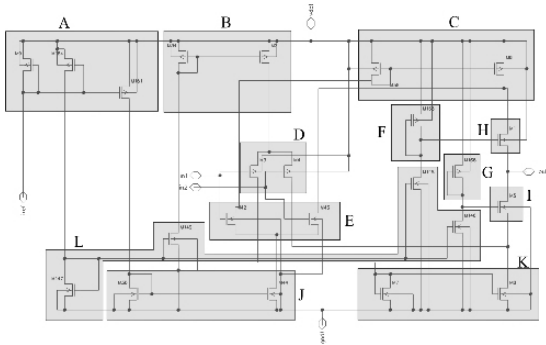


图 6 6 个实验电路对于 BMGA3和 SA 的运行指标对比图

图 7(a)给出了电路 1 的原理图,电路分块如图中阴影矩形框所示.图 7(b)给出了运行 BMGA3 所获得的布局结果,其质量可与手工布局相比拟.模块

的相对位置保存在 DA 中,将其输入布线器后,压缩所有模块并进行细节布线(detailed routing),最后完成整个版图的设计.



(a) 原理图

(b) 布局版图

图 7 杆对杆运算放大器的原理图和 BMGA3 的布局版图

### 5 结束语

### 参 考 文 献

本文介绍了一种用遗传算法解决模拟集成电路模块布局问题的新方法.它使用二维位矩阵作为染色体编码并配合灵活的遗传算子,模块滑行有机地融合了绝对布局和相对布局的优点,极大地提高了搜索的质量和效率,个体间相似性检查避免了算法的早熟收敛,目标函数包含了模拟集成电路的特殊要求.并采用正交实验的方法研究了算法的参数,一个衍化遗传算法用来确定最优的参数取值,该方法集成在 ALADIN 的设计助手中.实验结果表明,具有优化参数的该算法提高了设计效率,其布局质量可与手工布局相比拟.

- 1 Croes K, Man H J. A process-tolerant symbolic layout system. *IEEE Journal of Solid-State Circuits*, 1988, 23(3): 705~713
- 2 Zhang L, Kleine U, Roewer F *et al.* A novel design tool for analog integrated circuits. In: *Proceedings of First Joint Symposium on Opto- & Microelectronic Device and Circuits*, Nanjing, 2000. 146~149
- 3 Cheng C, Kuh E. Module placement based on resistive network optimization. *IEEE Transactions on Computer-Aided Design CAD-3*, 1984, 3(7): 218~225
- 4 Sechen C, Sangiovanni-vincentelli A. The timberwolf placement and routing package. *IEEE Journal of Solid-State Circuits*, 1985, sc-20(2): 510~522
- 5 Wong D F, Liu C L. A new algorithm for floorplan design. In: *Proceedings of the 23rd ACM/IEEE Design Automation Conference*, Berlin, 1986. 101~107
- 6 Cohn J M, Garrod D J, Rutenbar R A *et al.* KOAN/ANAGRAM II: New tools for device-level analog placement and routing. *IEEE Journal of Solid-State Circuits*, 1991, 26(1): 330~342
- 7 Rijmenants J, Litsios J B, Schwarz T R *et al.* ILAC: An auto-

致谢 作者们感谢德国 ATMEL 无线电和微型控制器公司的 Wolf 博士和法国 ISEN 研究所的 Berhelin 先生富有价值的讨论和建议.该课题在德国马格德堡大学完成,其资金支持来源于撒克森—安亥州和西门子公司.

- mated layout tool for analog CMOS circuits. *IEEE Journal of Solid-State Circuits*, 1989, 24(2): 417~425
- 8 Shahookar K, Mazumder P. A genetic approach to standard cell placement using meta-genetic parameter optimization. *IEEE Transactions on Computer-Aided Design*, 1990, 9(5): 500~511
  - 9 Chan H, Mazumder P, Shahookar K. Macro-cell and module placement by genetic adaptive search with bitmap-represented chromosome. *Integration, the VLSI Journal*, 1991, 28(9): 49~77
  - 10 Esbensen H, Mazumder P. SAGA: A unification of the genetic algorithm with simulated annealing and its application to macro-cell placement. In: *Proceedings of the 7th International Conference on VLSI Design*, London, 1994. 211~214
  - 11 Wolf M, Kleine U, Schafer F. A novel design assistant for analog circuits. In: *Proceedings of Asia and South Pacific Design Automation Conference*, Tokyo, 1998. 495~500
  - 12 Rohatgi V K. *An Introduction to Probability Theory and Mathematical Statistics*. Now York: John Wiley & Sons, 1976



**ZHANG Li-Hong**, born in 1971. Currently He is doing post-doctoral research work. His research interests are layout optimization and VLSI computer-aided design tools.

**XIE Chang-Sheng**, born in 1957, Ph. D. , professor and Ph. D. supervisor of the Department of Computer Science and Technology, HUST. His current research interests include computer storage system theory & technology, data storage technology and network multi-media storage technology.

**ZHANG Yu-Ping**, born in 1963, associate professor,

Ph. D. candidate at Shanghai Jiaotong University. Her current research interests include CAD and optimization technique.

**PEI Xian-Deng**, born in 1933, Ph. D. , professor and Ph. D. supervisor of the Department of Computer Science and Technology, HUST. His current research interests include computer storage system theory & technology, data storage technology and network multi-media storage technology.

**KLEINE Ulrich**, born in 1954, Ph. D. , professor and Ph. D. supervisor of the Department of Electrical Engineering, University of Magdeburg, Germany. His current research interests include analog and digital circuit design and design automation.