VLSI 冗余单元最优分配的遗传算法求解 [1]

赵天绪　　郝　跃　　周水生

(西安电子科技大学微电子所　西安　710071)

摘　要　随着 VLSI 芯片面积的增加和电路复杂性的增强，芯片的成品率受制造缺陷影响的概率逐渐增加. 为了解决这一问题，人们将容错技术结合入集成电路设计中. 要使一个系统具有较强的容错能力，必须给系统提供一定量冗余单元. 本文利用遗传算法有效地解决了使系统成品率达到最大的冗余单元最优分配问题.

关键词　遗传算法，成品率，备用单元，容错能力

中图号　TN47, TN911.72

1 引　言

本世纪 70 年代以来，集成电路方面取得的最大进步就是圆片面积的增加和器件的尺寸的微细化。然而，由于系统集成 (system on a chip) 的发展和迫切要求，人们把注意力集中到了大面积芯片甚至整个圆片的集成电路 (Wafer Scale Integration，即 WSI)。然而，大面积的 VLSI 电路对制造过程中的缺陷非常敏感，导致集成电路的成品率低下。为了使集成电路有一个可行的成品率，这些芯片必须设计成具有一定的容错能力。因此，容错技术又成为至今系统集成技术研究的热点。要系统具有容错能力就必须给系统中加入一定量的冗余单元，当集成电路系统中的处理单元出现故障时用这些冗余单元来替换出现故障的单元。人们最关心的问题是在系统中如何最优地分配加进的这些冗余单元才能使系统获得最优的成品率。 Thibeault[1] 等利用品质因数这个指标来估计使系统的品质因数达到最大时所需冗余单元的个数，该算法是一种在解存在的情况下在多项式时间内可能找不到最优解的启发式算法。 Shi[2] 等考虑了将一定量的冗余单元分配给 k 个子系统使得这 k 个子系统是好的概率最大，从而使系统的成品率达到最大。然而，对于该问题如何求出最优解，至今没有给出可行的算法。对本问题而言，由于目标函数的复杂性和变量的离散性，用传统的整数规划方法来搜索这个问题的全局最优解已显得不是十分有效，同时该问题也向传统的优化理论提出了严峻的挑战。

本文所用的遗传算法 (GA) 是一种模拟生物进化过程的基于模式理论的自适应优化技术。作为一种启发式的全局搜索算法，面向的是具有多局部极值的含有噪声的大空间搜索问题。它具有通用性、鲁棒性及隐并行计算性等优点。算法通过维持一个群体，并按照每个个体的适应度大小重复地进行交叉、遗传及变异操作来实现个体结构的优化重组，并最终达到或接近全局最优解。本文利用遗传算法有效地解决了将 m 个相同单元最优地分配给 k 个子系统，使得 k 个子系统都是好的概率达到了最大，从而保证了系统具有最大的成品率的问题。模拟结果表明遗传算法用于解决此类问题是十分有效的。

2 冗余单元最优分配模型

设一个系统由 k 个可完成不同功能的功能块组成，这些功能块由同一种处理单元组成 (如门阵列、标准单元、 ROM ， RAM ， PLA 等，它也可以是模拟电路单元)，只是构成每一个功能块的处理单元数和互连拓扑结构不同而已，即组成整个系统的基本单元是相同的。若每一个功能块处于正常工作状态，则系统正常工作；否则系统处于失效状态。为了使系统具有容错能力，必须给系统提供冗余单元。对加入冗余单元后的功能块，如第 i 个功能块，由 m_i 个处理单元组成，要使该功能块正常工作，至少要有 $n_i(n_i \le m_i)$ 个单元正常工

[1] 1999-05-09 收到， 1999-10-13 定稿
国家部级基金和 863 高科技基金资助项目

作，其余的 $m_i - n_i$ 个单元作为备用单元或者处于故障状态。此处仅考虑处理单元可能出现故障，而认为电路中的连线上无故障。那么，第 i 个功能块的成品率 $y(n_i, m_i)$ 为

$$y(n_i,m_i) = P_r(m_i\text{个单元中至少有 } n_i \text{ 个正常工作}) = \sum_{k_i=n_i}^{m_i} \binom{m_i}{k_i} p^{m_i-k_i}(1-p)^{k_i}$$

上式中 $P_r(\cdot)$ 为概率；p 为每一个单元成为故障单元的概率。由于每个功能块独立完成一定的功能，因此由 k 个功能块组成的系统正常工作的概率 (即系统的成品率)$y(n_1, m_1; n_2, m_2; \cdots; n_k, m_k)$ 为

$$y(n_1,m_1;n_2,m_2;\cdots;n_k,m_k) = \prod_{i=1}^{k} y(n_i,m_i)$$

若该系统中共有 m 个相同处理单元，最优冗余分配就是把正整数 m 分解成一组正整数 $(m_1, m_2, \cdots, m_k)(n_i \le m_i)$，且 $\sum_{i=1}^{k} m_i = m$，按照这种分解方式将处理单元的个数分配到相应的功能块中，使得整个系统的成品率 $y(n_1, m_1; n_2, m_2; \cdots, n_k, m_k)$ 达到最大。该问题的模型为

$$\left.\begin{aligned} &\max\ y(n_1,m_1;n_2,m_2,\cdots,n_k,m_k) \\ &\text{s.t.}\quad m_1+m_2+\cdots+m_k = m \\ &\qquad\quad n_i \le m_i,\ i=1,2,\cdots,k \end{aligned}\right\} \tag{1}$$

显然，这是一个有约束的整数规划问题。

3 用遗传算法实现冗余单元的最优分配

(1) 式表示的模型是一个有约束的非线性整数规划，它是 NP 完全问题。传统的优化算法难以保证求出该问题的全局最优解。遗传算法 (GA) 是一种模拟生物进化过程的全局随机搜索算法，它适用于求解传统搜索方法难以解决的复杂的优化问题。遗传算法是一种群体型操作，它以群体中的所有个体为对象，按照个体的适应度大小依一定的概率反复地进行选择、交叉和变异操作来实现个体结构的优化重组，直到接近或获得全局最优解为止。本文采用的遗传算法的程序结构如下：

```
procedure of GA
begin                                  /* 开始 */
t → 0 ;                                /* 初始化遗传代数 */
initialize M(t) ;                      /* 初始化群体 M(0)*/
repeat                                 /* 进入迭代循环过程 */
{calculate G(t) ;                      /* 计算第 t 代各个体的适应度 */
select M(t+1)from M(t) ;               /* 从第 t 代中选择第 t+1 代 */
crossover ;                            /* 交叉操作 */
mutate ;                               /* 变异操作 */
t ← t+1 ;                              /* 更新遗传代数 */
}until(stop-criterion)                 /* 如果满足终止条件就终止循环 */
end                                    /* 结束 */
```

(1) *编码和搜索空间的确定* 编码就是建立个体的数据结构。它把算法中的个体与问题的可行解对应起来。本文直接把可行解向量 $(m_1, m_2, \cdots, m_k)$ 作为个体的数据结构，且利用它们的二进制数串作为染色体代码。根据给定的约束条件分别确定出 m_i 的上限 m_{u_i} 和下限 $m_{l_i}(i = 1, 2, \cdots, k)$，即确定了遗传算法搜索的边界：$m_{u_i} = m - \sum_{j=1,j\ne i}^{k} n_j,\ m_{l_i} = n_i$ 。

(2) 适应度函数　一般的，可以把优化问题的目标函数直接作为适应度函数。然而，冗余单元的最优分配问题是一个有约束的优化问题，因此采用下面的方法来构造该问题的适应度函数：

$$G(n_1,m_1;n_2,m_2;\cdots;n_k,m_k)=\begin{cases} y(n_1,m_1;n_2,m_2;\cdots;n_k,m_k)+1, \\ \qquad 当 m_1+m_2+\cdots+m_k-m=0 \\ 1, \qquad 当 m_1+m_2+\cdots+m_k-m\neq 0 \end{cases}$$

这是因为当 $m_i(i=1,2,\cdots,k)$ 满足约束条件时，有 $0\leq y(n_1,m_1;n_2,m_2;\cdots;n_k,m_k)\leq 1$ 。为了保证求解过程中适应度始终为正，故加 1 ；而当 $m_i(i=1,2,\cdots,k)$ 不满足约束条件时，适应度取为 1 。

(3) 选择　选择操作的目的是从当前群体中选出优良的个体，使之有机会作为父代繁殖下一代。判断个体优劣与否的准则是各自的适应度值。显然这一操作是借用了 Darwin 的“适者生存”的进化原则，即个体适应度越高，其被选择的机会就越多。被选择的个体数目由选择概率 P_s 控制。选择操作的实现方式很多，本文采用的是 Roulette wheel[3] 选择方式。

(4) 交叉　所谓交叉是指把两个父代个体的部分结构加以替换重组而生成新个体的操作。通过交叉，遗传算法的搜索能力得以大大提高。对随机选定的两个个体，本文将采用一致 (uniform) 交叉 [4] 来产生两个新的个体。一致交叉的概率用 P_c 来表示，一般取 P_c 从 0.1 到 0.9 之间。

(5) 变异　在遗传算法中引入变异的目的有两个：一是使它具有局部的随机搜索能力，二是使它维持群体的多样性，防止出现未成熟收敛现象，以避免迭代陷入局部极值解。本文采用的变异操作为，对群体中的个体二进制码串随机挑选一个或多个基因座，并对这些基因座以概率 P_m 进行变异操作 [4] ，即从 0 变到 1 ，从 1 变到 0 。一般取 P_m 从 0.01 到 0.05 之间。

(6) 迭代终止条件　本文采用的终止条件为进化代数 T 。遗传算法在计算了进化代数 T 后终止。

4 试验结果与分析

为了说明遗传算法对解决备用单元最优分配问题的有效性，本文给出了一些数据并作了数值试验。在本文中，遗传算法的参数设定为 $M=150$, $P_r=0.8$, $P_c=0.2$, $P_m=0.01$ 。算法采用的终止条件是进化代数 T 。分别进行不同的试验，其结果如下：

试验 1　$p=0.1; n_1=n_2=n_3=n_4=20; T=200$. 由组合理论知该问题的最优解是给每一功能块分配 25 个单元。利用遗传算法搜索得到的最佳个体为 $m_1^*=m_2^*=m_3^*=m_4^*=25$ ，系统的最大成品率 $Y^*=0.872946$ ，与理论结果完全符合。

试验 2　$p=0.1; m=1000; n_1=100, n_2=200, n_3=240, n_4=300; T=500$. 遗传算法搜索到的最佳个体为： $m_1^*=124, m_2^*=239, m_3^*=284, m_4^*=353$ ；系统的最大成品率 $Y^*=0.99907$ 。此例说明当 m 较大时，遗传算法也能给出全局最优解。

图 1 和图 2 分别表示试验 1 及 2 的收敛过程，从图中可以看出遗传算法的收敛速度很快。图 1 表明，在第 120 代时就使系统的成品率达到最大；图 2 表明，在第 320 代时就使系统的成品率达到最大。大量算例均说明利用遗传算法解决此类问题是十分有效的。

5 结论

利用遗传算法有效地解决了有容错的 VLSI 的备用单元的最优分配问题。大量数值试验表明此算法用于解决该类问题收敛速度快，而且可以给出全局最优解。因此，遗传算法可用于求解传统的优化方法难于解决的问题，同时也为解决 VLSI 系统集成容错设计问题开辟了一条非常有效的途径。

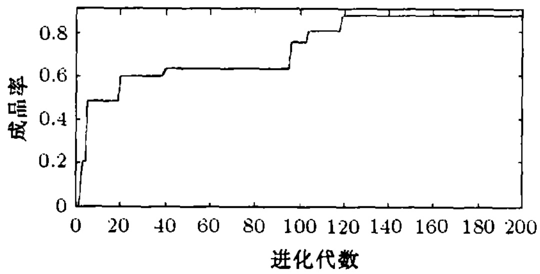

图 1　GA 对试验 1 的收敛过程

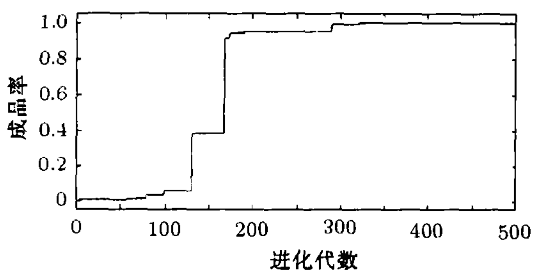

图 2　GA 对试验 2 的收敛过程

参 考 文 献

[1] C. Thibeault, Y. Savaria, J-L. Houle, A fast method to evaluate the optimum number of spares in defect-tolerant integrated circuits, IEEE Trans. on Comput., 1994, C-43(6), 687–697.

[2] W. Shi, W. K. Fuchs, Optimal spare allocation for defect-tolerant VLSI, in Proc. IEEE Int. Conf. Wafer Scale Integration, San Francisco, California, USA, 1992, 193–199.

[3] 云庆夏，黄光球，王战权，遗传算法和遗传规划，北京，冶金工业出版社，1997, 21–38.

[4] D. E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Reading, MA, Addsion-Wesley, 1989, 97–108.

USING THE GENETIC ALGORITHMS TO SOLVE THE PROBLEM OF OPTIMAL SPARE ALLOCATION FOR FAULT-TOLERANT VLSI

Zhao Tianxu　　Hao Yue　　Zhou Shuisheng

(*Research Inst. of Microelectronics, Xidian University, Xi'an* 710071, China)

Abstract　An increase in chip area and circuit complexity leads to a reduction in the yield of production. In order to solve the problem of low yield by defects in process of large scale integrated circuits manufacture, the fault-tolerant technique is introduced into the integrated circuits design. A system must be provided with a certain quantity of spare elements to have the ability of fault-tolerance. In this paper, the problem of optimal spare allocation for fault-tolerant VLSI is solved effectively by means of genetic algorithm.

Key words　Genetic algorithm, Yield, Spare element, Fault-tolerant ability

赵天绪：　男，1964 年生，副教授，博士，主要从事 VLSI 的可制造性、IC 容错结构分析及成品率建模方面的研究.

郝　跃：　男，1958 年生，教授，博士生导师，主要从事 VLSI 的可制造性工程与设计方法学、IC 统计模型与优化、IC 可靠性模型与设计以及新型器件与电路研究.

周水生：　男，1972 年生，讲师，主要从事多层优化、多目标规划和遗传规划的理论及算法研究等.