

基于 SPEA-II 算法的 SoC 测试多目标优化研究*

谈恩民 朱 峰 尚玉玲

(桂林电子科技大学电子工程与自动化学院 桂林 541004)

摘要: 测试功耗、测试时间是 SoC 测试优化中的两个测试目标,它们之间存在相互影响的关系。在多目标优化过程中,进化算法对于解决多目标优化问题拥有比较好的优化效果,因此各种进化算法被广泛地应用于 SoC 测试多目标优化的研究中。对 SoC 测试时间、测试功耗这两个测试目标建立联合优化模型,分析了 NSGA-II 算法与 SPEA-II 算法的特点,并对改进型强度 Pareto 进化算法(SPEA-II)进行研究,进而将 SPEA-II 算法用于上述所建立模型的求解。使用 ITC'02 标准电路中 p93791 电路和 d695 电路对上述方法进行实验验证,实验数据表明上述方法可以求得该联合优化模型的一组最优解;并且针对 p93791 电路,在与 NSGA-II 算法的实验数据比较中,得到了更好的优化结果。证明了 SPEA-II 算法对 SoC 测试结构优化方面具有良好的适用性和可行性。

关键词: SPEA-II 算法; SoC 测试; 测试时间; 测试功耗

中图分类号: TP302 **文献标识码:** A **国家标准学科分类代码:** 520.1040

Optimization of SoC test multiple objects based on SPEA-II algorithm

Tan Enmin Zhu Feng Shang Yuling

(Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: In the system chip SoC test optimization, test power and test time are the two test target, they exist interaction relations. In the process of multi-objective optimization, evolutionary algorithm for solving multi-objective optimization problem has good effect, so various evolutionary algorithms are widely used in SoC test in the study of multi-objective optimization. In the absence of SoC test time or test power consumption as the constraint conditions under the premise of the SoC test time and test power joint optimization model of the these two goals. And the improved Strength Pareto Evolutionary Algorithm (SPEA-II) is studied, which will SPEA-II Algorithm used for the solution of the model established. Use the ITC'02 standard p93791 circuit and d695 in the circuit of the above methods to experimental verification, the results show that the method can equilibrium solution is provided for the established joint optimization model. And in view of the p93791 circuit, in the experimental data with the NSGA-II algorithm comparison, better optimization results were obtained. To prove to SPEA-II SoC test architecture optimization has good applicability and effectiveness.

Keywords: SPEA-II ; SoC test; test time; test power

1 引言

随着时代发展,SoC 功能越来越复杂,SoC 内部集成的 IP 核种类越来越多,数量也越来越大,使得对 SoC 测试成本影响严重的因素——测试时间急剧增加。怎样降低 SoC 的测试时间从而达到降低其测试费用成为 SoC 系统级测试中急需解决的重要问题^[1]。测试时间、测试功耗,以及测试电路面积等因素是测试优化中需要考虑的重

要因素。然而这些因素相互间存在着制约关系,要使这些因素都达到最优化状态是不切实际的。

目前对于 SoC 测试结构的优化研究,国内以及国外的学者大部分将精力集中于单一因素的优化上。部分学者也仅仅是在研究测试时间的优化问题时,将测试功耗作为约束条件^[2-4]。文献[5-6]中分别建立了对测试时间与测试功耗联合优化的优化模型,并采用不同的进化算法进行求解。所使用的算法能够快速获得最优解集,但其解集

收稿日期:2015-03

* 基金项目:国家自然科学基金(61102012)、广西自动检测技术与仪器重点实验室主任基金(YQ14105)项目

分布均匀性不强,算法容易过早收敛,陷入局部最优解,丢失全局最优解。

2 SoC 测试的多目标优化问题的描述

本文所研究的优化模型为:对于已定的 SoC 芯片,IP 核的个数为 N 个,测试总线的宽度为 W ,主要对基于 IP 核的 SoC 测试时间以及测试功耗这两个目标进行同时优化。对于两个或者两个以上优化目标的优化问题,为多目标优化问题^[7-8]。对于含有 N 个 IP 核的 SoC 芯片,如果对所有的 IP 核进行并行测试,则总的测试时间会得到最大程度的减少,可是却会导致瞬时测试功耗达到最大值;如果对所有的 IP 核进行串行测试,则瞬时测试功耗会得到最大程度的降低,但是总的测试时间却达到最大。故采用并串测试相结合的方式对 SoC 芯片进行测试,将 N 个 IP 核分成 k 组进行测试,每组测试的 IP 核数目为 m_k 个。同一组内的各个 IP 核使用并行测试的方法,不同组之间的 IP 核使用串行测试的方法。采用以下目标函数求解。

1)最小化 SoC 的整体测试时间。总的测试时间目标函数可以表示为:

$$F(1) = \text{Min} \left\{ \sum_{i=1}^k \max \{ T_1 \cdot y_{i1}, \dots, T_j \cdot y_{ij}, \dots, T_{N \cdot y_{ik}} \} \right\} \quad (1)$$

式中: T_j 为第 j 个 IP 核的测试时间,其中 $j \in [1, N]$ 。 y_{ij} 为第 i 组第 j 个 IP 核的状态,其中 $i \in [1, N]$,即: $y_{ij} = 1$,表示 IP 核为测试状态; $y_{ij} = 0$,表示 IP 核为空闲状态。

2)因为动态功耗是 CMOS 数字电路中占比最大的功耗,所以主要考虑 SoC 测试过程中产生的动态功耗。SoC 测试的瞬时总功耗的目标函数表达式为:

$$F(2) = \text{Min} \left\{ \text{Max} \left\{ \sum_{j=1}^N P_j \cdot y_{ij} \right\} \right\} \quad (2)$$

式中: P_j 为测试 IP 核 j 所产生的瞬时功耗; y_{ij} 与式(1)中的取值相同。

3)约束条件:

$$\sum_{i=1}^k m_i = N; \sum_{j=1}^N y_{ij} = m_i; y_{ij} \neq y_{i+1j} \quad (3)$$

式中: i, j, k 和 y_{ij} 与式(1)中的取值相同。

3 SPEA-II 算法

由于所研究的是测试时间与测试功耗协同优化问题,遗传算法是一个全局寻优的算法,寻优结果综合各个目标考虑,是最优解^[9]。而在众多的算法中,蚁群优化算法^[10]、NSGA-II 算法和 SPEA-II 算法都是比较优秀的多目标进化算法。

NSGA-II 算法的优点在于运行效率高、解集具有良好的分布性,对于低维度优化问题具有良好的表现;其缺点在于对高维度优化问题的求解过程存在缺陷,解集的多样性不够理想。

SPEA-II 算法是 Zitzler 和 Thiele 于 2001 年,在 SPEA

算法的基础上提出的改进算法。SPEA-II 算法在适应度赋值方法方面得到了改进;并且采用了新的环境选择策略,新的环境选择策略基于邻近规则的特点,使得 SPEA-II 算法所获得的解集具有优于其他算法的分布均匀性,特别在对高维度优化问题的求解上。

虽然 SPEA-II 算法存在运行效率不高的缺点,但在面对 SoC 快速发展、SoC 测试复杂程度越来越高的发展趋势,SPEA-II 算法比 NSGA-II 算法更能应对高维度优化问题,所以本文选择 SPEA-II 算法作为研究对象。

3.1 算法主要流程

图 1 所示内容为 SPEA-II 算法的主要运算流程。

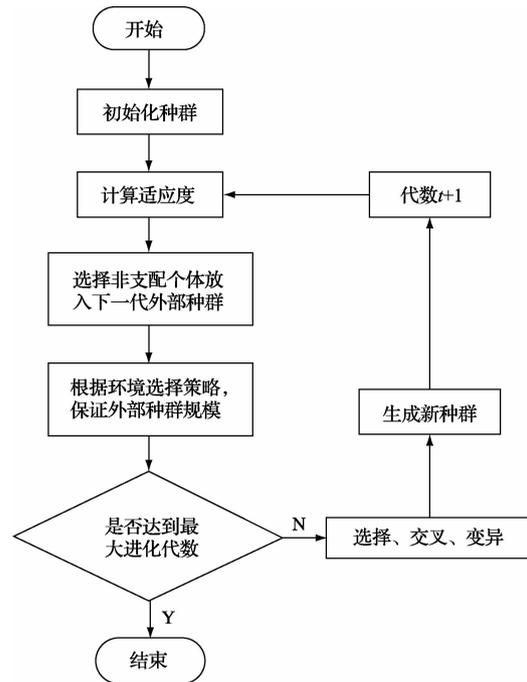


图 1 SPEA-II 算法主运算流程

3.2 SPEA-II 算法的主要核心思想

SPEA-II 算法采用了将细粒度的原始适应度赋值法和密度估计法这两种方法相结合的新的适应度赋值法。对于每个个体,细粒度的适应度赋值法不仅考虑了它们的支配情况,而且还考虑了它们的被支配情况。设个体 i 的个体强度为 $S(i)$,计算方法如式(4):

$$S(i) = |\{j | j \in P_i \cup E_i, \text{且 } i < j\}| \quad (4)$$

设个体 i 的原始适应度为 $R(i)$,计算方法如式(5):

$$R(i) = \sum_{j \in P_i \cup E_i, j < i} S(j) \quad (5)$$

当 $R(i) > 0$ 的时候,个体 i 为支配个体, $R(i)$ 的值越小表示个体 i 的支配等级越高;当 $R(i) = 0$ 的时候,个体 i 为非支配个体。当 $R(i) = R(j)$ 时,个体 i 与个体 j 之间的差异通过密度估计法进一步区分。SPEA-II 算法采用的密度估计方法为 k -th 最近邻域法 (k -nearest neighbor techniques),计算方法如下:

首先计算个体 i 到 P_i 和 E_i 中除了个体 i 以外,其余每一个个体的距离,然后按照升序排列:

$$d_1(i) \leq d_2(i) \leq \dots \leq d_n(i)。$$

密度估计公式为:

$$\delta_i^* = k/[2ndk(i)] \quad (6)$$

式中: $k \approx \sqrt{N+M}$, N 表示当代进化种群 P_i 的种群大小, M 表示外部种群 E_i 的种群大小。

个体 i 的密度 $D(i)$ 的定义如式:

$$D(i) = 1/(\delta_i^* + 2), (0 < D(i) < 1) \quad (7)$$

个体 i 的适应度 $F(i)$ 定义如式:

$$F(i) = R(i) + D(i) \quad (8)$$

由式(8)可知,当个体 i 是非支配个体时, $F(i) < 1$ 。

SPEA-II 算法的环境选择策略:从当代进化种群 P_i 和当代外部种群 E_i 中选择非支配个体,将其保存在下一代外部种群 E_{t+1} 中。若当代进化种群 P_i 和当代外部种群 E_i 中非支配个体的数量小于外部种群的大小时, SPEA-II 算法的环境选择策略将在当代进化种群 P_i 和当代外部种群 E_i 中继续选择一些支配等级较高的支配个体,将其保存在下一代外部种群 E_{t+1} 中,以维持解集的多样性。若当代进化种群 P_i 和当代外部种群 E_i 中非支配个体的数量大于外部种群的大小时, SPEA-II 算法的环境选择策略采用基于邻近规则的截断策略,不仅能够保持外部种群的大小不变,而且保证能够边界个体不被去除。

环境选择策略的具体过程如下:

步骤 1: 设 P_t 为进化种群, E_t 为外部种群。 N 表示种群 P_t 的种群大小, M 表示外部种群 E_t 的种群大小。将非支配个体保存在下一代外部种群中,即:

$$E_{t+1} = \{i | i \in P_t \cup E_t, \text{且 } F(i) < 1\} \quad (9)$$

步骤 2: 当 $|E_{t+1}| = M$, 则环境选择过程结束。否则转到步骤 3。

步骤 3: 当 $|E_{t+1}| < M$, 则将 E_t 和 P_t 中支配个体依照支配等级进行降序排列,并选取前 $M - |E_{t+1}|$ 个支配个体存入 E_{t+1} 中,环境选择过程结束。否则转入步骤 4。

步骤 4: 当 $|E_{t+1}| > M$, 则使用截断策略去除多余个体,环境选择过程结束。

截断策略的具体方法:定义算子 $i \leq d_j$ 为:

$$\left\{ \begin{array}{l} \forall 0 < k < |E_{t+1}| : \delta_i^k = \delta_j^k \vee \\ \exists 0 < k < |E_{t+1}| : \left[\begin{array}{l} (\forall 0 < l < k : \delta_i^l = \delta_j^l) \\ \wedge (\delta_i^k < \delta_j^k) \end{array} \right] \end{array} \right. \quad (10)$$

当个体 i 对 E_{t+1} 中其他每一个个体满足 $i \leq d_j$ 时,将个体 i 移出 E_{t+1} 。

4 基于 SPEA-II 算法的测试优化

4.1 整数向量编码

本文采用整数编码方案。记整数向量: $X = (x_1, x_2, \dots, x_i, \dots, x_N)$ 表示 SoC 内部 N 个 IP 核的一种测试方

案,其中 $x_i \in [1, N]$ 的任意整数。若 x_i 与 x_j 取值相同,则表示第 i 个 IP 核模块与第 j 个 IP 核模块属于同一组。

将 X 转化为解 Y , Y 中的 x_{ij} 为二进制编码, $j \in [1, N]$:

$$Y = (k, m_i, x_{i1}, x_{i2}, x_{i3}, \dots, x_{iN})$$

4.2 IP 核测试分配的算法实现

设进化种群大小为 N , 外部种群大小为 M , 当前进化代数数为 t , 最大进化代数数为 T 。

步骤 1: 初始化种群 P_t , 外部种群 E_t , $t = 0$ 。

步骤 2: 使用环境选择法,生成下一代外部种群 E_{t+1} 。

步骤 3: 如果 $t < T$, 执行步骤 4。否则输出 E_{t+1} 中的非支配个体。

步骤 4: 从 E_{t+1} 中选择 N 个个体,进行遗传操作,产生下一代种群 P_{t+1} 。

步骤 5: $t = t + 1$, 执行步骤 2。

5 IP 核测试分配的应用实例

运用 SPEA-II 算法,对 IP 测试分配进行求解,使其达到 SoC 测试时间与测试功耗的联合优化效果。采用国际标准 SoC 电路 IT'02 SoC Test Benchmark 来进行分析、验证 SPEA-II 算法的适用性和有效性。分别采用了标准电路中较为复杂的 p93791 电路,以及 IP 核模块较少的 d695 电路对算法进行实验验证。

5.1 p93791 电路应用实例

表 1 中,已知 TAM 总线宽度为 18, p93791 电路的各个 IP 核模块的测试时间通过 BFD 算法计算获得。

表 1 p93791 各 IP 核模块测试时间和测试功耗

IP 核	测试时间 / μ s	测试功耗 /mW	IP 核	测试时间 / μ s	测试功耗 /mW
1	185 319	7 014	17	96 998	6 674
2	770	74	18	254	113
3	2 594	69	19	62 666	5 253
4	154	225	20	215 588	7 670
5	67 406	248	21	254	113
6	341 858	6 150	22	170	76
7	532	41	23	120 554	7 844
8	532	41	24	6 145	21
9	770	77	25	8 065	45
10	18 632	395	26	386	76
11	15 603	862	27	186 150	3 135
12	109 367	4 634	28	3 171	159
13	123 239	9 741	29	64 874	6 756
14	123 239	9 741	30	770	77
15	1 154	78	31	2 044	218
16	3 568	201	32	49 352	396

设置初始进化种群大小为 300,交叉概率为 0.8,变异

概率为 0.1,最大进化代数 为 100。算法使用 MATLAB 编程实现。图 2 所示为 p93791 电路采用 SPEA-II 算法所得到 Pareto 最优解分布图。图中的点为目标函数 $F(1)$ 和 $F(2)$ 的非支配前段个体。加方框标示,坐标为(1 334 335, 10 805)的点为该优化结果的无偏最优解。 $F(1)$ 和 $F(2)$ 分别为 p93791 电路总的测试时间和测试功耗,单位分别是 μs 和 mW 。

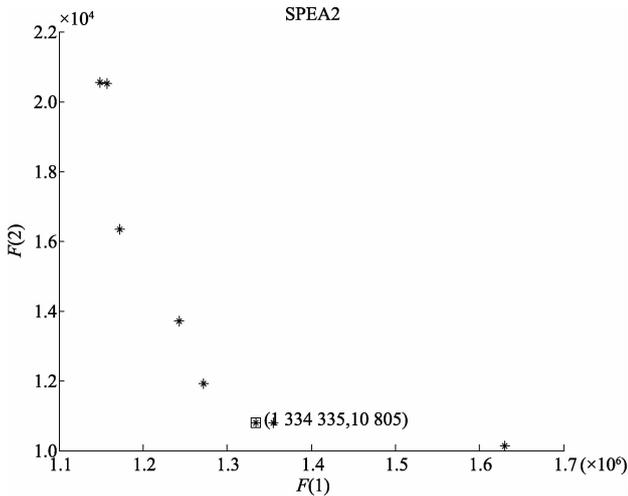


图 2 p93791 电路 SPEA-II 算法优化结果

根据 SPEA-II 算法的计算结果,表 2 列出了部分个体。

表 2 部分 Pareto 解

IP 核	编码 1	编码 2	编码 3	编码 4	IP 核	编码 1	编码 2	编码 3	编码 4
1	17	21	25	21	17	28	23	4	23
2	15	26	10	26	18	28	6	4	19
3	18	21	4	21	19	19	23	23	6
4	18	11	14	11	20	13	10	25	10
5	28	25	4	25	21	19	21	10	21
6	13	17	25	17	22	28	21	19	21
7	18	21	23	21	23	28	11	4	11
8	28	25	4	25	24	28	21	14	21
9	15	11	30	11	25	18	11	23	11
10	28	21	4	21	26	19	25	14	25
11	19	11	23	11	27	13	10	4	10
12	18	17	23	17	28	18	11	4	11
13	18	26	14	26	29	19	4	23	19
14	17	25	14	25	30	15	6	4	19
15	28	11	14	11	31	28	21	4	21
16	19	21	14	21	32	19	11	4	11

p93791 电路所有种群个体的各 IP 核的编码都为

[1,32]的整数。其中编码转化解 Y 的形式后。

$k=6$,表示 32 个 IP 核模块分为 6 组进行测试, $m_1=2, m_2=3, m_3=7, m_4=10, m_5=3, m_6=7$,分别表示各组分分配的 IP 核的个数。

表 3 列出了表 2 中每个解对应的目标函数值。在表 3 中可以看出,经过 SPEA-II 算法优化后获得的最优解集分布均匀。

表 3 目标函数值

	总测试时间	总测试功耗
编码 1	836 614	16 955
编码 2	1 172 439	11 927
编码 3	762 324	20 834
编码 4	1 334 335	10 805

5.2 d695 电路应用实例

采用较为复杂的、IP 核模块较多的 p93791 电路进行验证实验之后,采用 IP 核模块相对较少的 d695 电路进行了同样的验证实验,以证明 SPEA-II 算法具有良好的适用范围。

表 4 列出了 d695 电路中各个 IP 核模块的测试时间和测试功耗。已知 TAM 总线宽度为 32,d695 电路的各个 IP 核模块的测试时间通过 BFD 算法计算获得。

表 4 d695 各 IP 核模块测试时间和测试功耗

IP 核	测试时间 (TAM=32)/ μs	测试功耗 /mW
1	25	660
2	588	602
3	5 058	823
4	6 888	275
5	6 207	690
6	12 213	354
7	4 314	530
8	5 778	753
9	836	641
10	4 136	1 144

对于 SPEA-II 算法的参数,设置初始种群大小为 100,交叉概率为 0.8,变异概率为 0.1,最大进化代数为 100。表 5 列出了该组 Pareto 最优测试方案解的部分解。

表 5 部分 Pareto 最优解方案及对应目标函数值

序号	编码方案	测试时间	测试功耗
1	2 2 8 8 7 8 5 7 5 1	27 458	1 452
2	9 5 6 10 10 6 6 10 9 5	24 073	1 746
3	9 9 5 3 3 3 3 5 7 7	22 715	1 849
4	2 6 9 7 7 7 9 1 2 4	19 415	2 463

6 SPEA-II 算法与 NSGA-II 算法的实验结果比较

部分实验数据如表 6 所示。

表 6 SPEA-II 与 NSGA-II 的实验结果对比

序号	SPEA-II		NSGA-II		SPEA-II 比 NSGA-II 改进效果	
	总的测试时间 / μs	总的测试功耗 /mW	总的测试时间 / μs	总的测试功耗 /mW		
1	836 614	16 955	866 486	17 585	3.4%	3.5%
2	1 172 439	11 927	1 203 165	12 893	2.5%	7.4%
3	762 324	20 834	765 511	23 606	0.4%	11.7%
4	1 334 335	10 805	1 349 171	11 201	1.1%	3.5%
5	1 057 082	13 820	1 086 207	13 820	2.6%	0.0%
6	714 886	22 638	758 834	23 668	5.7%	4.3%

根据文献[7]可得: NSGA-II 算法对 p93791 电路进行分析、优化,在 TAM=18 的前提下运行 NSGA-II 算法 100 代,得到的一组 Pareto 最优解。

从表 6 可知,在 TAM=18 的时, SPEA-II 算法对比 NSGA-II 算法在测试时间和测试功耗两个方面同时取得一定的改进。SPEA-II 算法和 NSGA-II 算法在执行相同迭代次数时, SPEA-II 算法的寻优能力更强。

7 结 论

本文通过建立测试功耗与测试时间联合优化的数学模型,采用分组并串测试的方法对基于 IP 核的 SoC 片上系统进行测试。运用 SPEA-II 算法对该数学模型进行求解。优化结果表明本文建立的数学模型以及采用的 SPEA-II 算法具有良好的有效性;从对比结果可以证明本算法具有良好的适用性。SPEA-II 算法对多目标优化问题的能够较好地避免丢失最优解,求得分布均匀的 Pareto 最优解集。

参 考 文 献

[1] 邓立宝,俞洋,彭喜元. 一种灵活 TAM 总线分配的 SoC 测试调度方法[J]. 仪器仪表学报,2011,32(6):

1238-1244.

- [2] 汪滢,王宏,李辛毅. 基于遗传算法的 SOC 测试功耗与测试时间协同优化[J]. 仪器仪表学报,2006,27(6):2327-2328.
- [3] 汪滢,许东宁. SOC 测试时间与测试功耗协同优化[J]. 微计算机信息,2009,25(11):27-29.
- [4] 赵丽丽,谈恩民,王海超. 优化 SOC 测试时间的扫描链平衡及 NSGA-II 设计[J]. 国外电子测量技术,2014,33(7):32-35.
- [5] 许川佩,胡红波. 基于量子粒子群算法的 SOC 测试调度优化研究[J]. 仪器仪表学报,2011,32(1):113-119.
- [6] 谈恩民,王鹏. 基于 NSGA-II 算法的 SOC 测试多目标优化研究[J]. 电子测量与仪器学报,2011,25(3):226-232.
- [7] 朱爱军,李智,许川佩. 三维 IP 核测试封装扫描链多目标优化设计[J]. 电子测量与仪器学报,2014,28(4):373-380.
- [8] KUNDU P K, ZHANG Y, RAY A K. Multi-objective optimization of simulated countercurrentmoving bed chromatographic reactor for oxidative coupling of methane [J]. Chemical Engineering Science, 2009, 64(19): 4137-4149.
- [9] 刘亚丽,马世伟. 基于遗传算法的列车自动运行系统的优化研究[J]. 电子测量技术,2013,36(4):36-39,43.
- [10] 崔小乐,熊志天,程伟,等. 热感知的 SoC 蚁群优化测试调度方法[J]. 仪器仪表学报,2014,35(4):948-953.

作 者 简 介

谈恩民,1967 年出生,教授。主要研究方向为计算机辅助测试和集成电路可测试性技术。

朱峰,1990 年出生,硕士研究生。主要研究方向为计算机辅助测试核集成电路可测试性设计。

尚玉玲,1977 年出生,博士,副研究员。主要研究方向为集成电路测试、信号完整性分析。