

一种基于遗传算法的多 IP 核并行测试方法

谈恩民, 贾亚平

(桂林电子科技大学 电子工程与自动化学院, 广西 桂林 541004)

摘 要: 针对现有 SoC 测试方法所需测试时间过长的问题, 提出了一种基于遗传算法的多 IP 核并行测试方法. 该方法主要是在功耗约束的情况下, 通过遗传算法将尽可能多的 IP 核的测试数据压缩, 即 IP 核的测试数据相同的位接到同一根总线数据位, 使每次并行测试的 IP 核数量达到最大. 此方法不仅可以减少测试数据集的大小, 且能够减少对总线位宽的需求. 通过应用遗传算法对所建立测试数据求取最佳测试方案的仿真, 证明了该方法是可行的, 且能够极大缩短测试时间.

关键词: SoC 测试; 遗传算法; IP 核; 并行测试

中图分类号: TN431

文献标识码: A

文章编号: 1000-7180(2017)01-0071-05

A Multiple IP Core Concurrent Test Method Based on Genetic Algorithm

TAN En-min, JIA Ya-ping

(School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract: Aiming at the problem of the existing SoC test methods required long testing time, this paper proposes a multiple IP core concurrent test method based on genetic algorithm. This method is mainly under the condition of the power consumption constraints, through the genetic algorithm as much as possible of the IP core test data compression. In other words, a bus access multiple IP core with the same test data, make each time the number of IP core of concurrent test to achieve the most. This method not only can reduce the size of the test data set, and can reduce the demand for bus bits wide. By using genetic algorithm to calculate the best test scheme of simulation show that the method is feasible, and can greatly shorten the test time.

Key words: SoC test; genetic algorithm; IP core; concurrent test

1 引言

随着集成电路的高速发展, 片上系统 (SoC) 中所集成的 IP 核越来越多, 功能越来越强大, 但是, IP 核的增加给 SoC 的测试带来了很大的困难. 基于 IP 核的 SoC 的测试方法主要有总线测试和边界扫描测试的方法. 对于基于总线的测试访问机制 (TAM) 结构而言, 当 IP 核的数量不是很多的时候, 可以通过 IP 核的测试调度, 使测试时间减少, 但是如果 IP 核数量过多, 完成测试所需要的时间仍然会很长; 而边界扫描的方法由于数据串行传输, 因此测试速度

慢, 且随着 IP 核的增加, 会造成测试控制器 (TAP) 的设计难度随之增大.

针对这些问题, 本文在总线测试方法的基础上提出了, 在功耗约束的情况下, 通过遗传算法将测试数据压缩到总线位宽范围内的测试方法, 即每次并行测试时使尽可能多的 IP 核接入总线. 同时, 为了避免由于 IP 核输入输出线与总线之间的线路连接问题而造成测试故障, 本文采用边界扫描完成数据线路的检测. 因为边界扫描方法不仅能够完成对 IP 核内部的测试, 而且能够进行系统级测试^[1].

2 测试原理

现有的 SoC 测试通常是通过优化 IP 核的测试调度从而实现多 IP 核的并行测试,最终达到缩短 SoC 测试时间的目的^[2].其测试调度效果如图 1 所示,其中图 1(a)显示的是未经过优化的调度,图 1(b)所示为优化后的调度,从图中可以看到优化后的测试时间减少了.

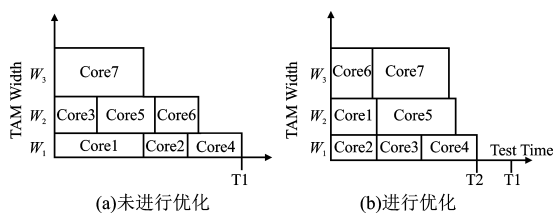


图 1 SoC 测试调度

为实现图 1 中的测试调度优化,也就出现了基于量子粒子群的 SoC 测试调度优化^[3]、基于遗传算法的 SoC 测试调度方法^[4-5]和基于 NSGA-II 算法的 SoC 测试多目标优化^[6]等方法.而文献^[7]所提出的灵活 TAM 总线分配的 SoC 测试调度进行了进一步的优化.虽然这些方法的确有效减少了测试时间,但是随着 IP 核的数量的增加,其测试时间仍然过长.而如果过多的增加总线位宽,则会造成布线困难、数据线之间容易相互干扰和测试数据过大等问题.因此,希望实现如图 2 所示的优化后的测试调度,其表示在功耗和总线位宽约束的情况下,尽可能使每次测试的 IP 核数量达到最大.如此,不仅能够解决 IP 核数量过多时总线位宽不够的问题,且能够极大地缩短测试时间.

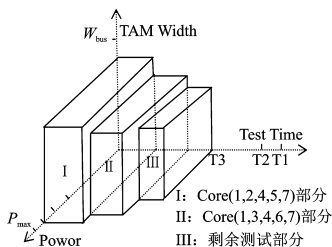


图 2 本文期望的 SoC 测试调度

图 2 表示在第 I 区时,对 IP 核 1、2、4、5、7 的部分数据进行测试.此时所测 IP 核的总功率在限定功耗范围内,而所有 IP 核的输入输出位宽之和肯定大于总线位宽 W_{bus} ,因此,需要将拥有相同测试数据的输入输出接入同一总线数据位.但是,由于 IP 核

的数量多,因此功耗满足情况下的每次测试选择也很多,所以如何确定 IP 核的测试顺序和 IP 核测试数据的输入顺序成为关键性问题.而由于遗传算法求解最优值的精度最高,因此,本文选择通过遗传算法来解决此问题.

3 测试系统结构

为完成基于遗传算法的多 IP 核并行测试,则必须要有与之相应的测试系统,因此,结合测试控制的需要,将测试系统结构设计为图 3 所示.其主要组成:输入总线、输出总线、测试控制总线、处理器 (CPU)、输入控制器、输出控制器、IP 核开启控制器、TAP 控制器及所有要测试的 IP 核.

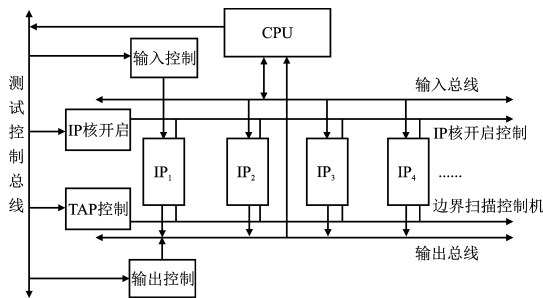


图 3 测试系统结构图

处理器负责整个测试的完成.先读取所有需要测试的 IP 核,再根据遗传算法及相应数据求取最佳测试方案,最后根据测试方案进行测试.输入控制器和输出控制器则是负责将测试 IP 核的输入输出线正确接入总线. IP 核开启控制器主要负责开启测试的 IP 核. TAP 控制器主要控制 IP 核的边界扫描单元完成 IP 核与处理器之间的连接故障检测,其具体设计参照文献^[8].由于每一根数据线只能够接收一位输入信号,所以对于 IP 核无故障时输出相同的位不能够直接接入同一总线位,必须先通过同或门或者异或门将其转换成一位数据.

该测试系统完成对所有 IP 核测试的步骤如下所示.

Step1: CPU 控制 TAP 与输入输出控制器,完成 IP 核与处理器之间的数据线路的检测;

Step2: CPU 通过总线读取所需要测试的 IP 核集 $V = (IP_1, IP_2, IP_3, \dots, IP_n)$;

Step3: 由测试 IP 核集从测试数据库中找到需要的测试数据集;

Step4: 根据测试的 IP 核集,确定族群的个体及

染色体长度,并选取初始族群;

Step5:进入遗传算法,求取最佳测试方案数据;

Step6:根据最佳测试方案数据完成 IP 核的测试,并且根据校对集判别否有故障.

在以上步骤中,遗传算法是该测试方法的核心,而由遗传算法求取的测试方案数据的优劣对测试时间的减少起决定性作用.因此,下文主要探讨遗传算法的实现和应用遗传算法求取测试方案数据的仿真验证.

4 遗传算法的实现

在本测试方法中,遗传算法主要任务是找到哪些 IP 核接入总线时,能够最佳的利用好总线的每一位,且根据最优的方法计算出此方法所形成的新的测试方案数据.因此,遗传算法的伪代码如下.

```
Genetic_Algorithm
{
    确定初始族群中个体编码的范围已及染色体长度;
    while(未完成所有 IP 核的测试)
        if(族群进行了更新)
            重新确定族群个体及染色体长度;
            从族群中随机选取初始族群;
        for(遗传迭代次数未完成)
            if(所有 IP 核的功率满足要求)
                求取测试方案数据集并且跳出 for 循环;
            计算个体适应度并按照适应度大小排序;
            if(本次种群最佳适应度比前种群大)
                更新测试方案数据集与最佳适应度;
                选择操作;
                交叉操作;
                变异操作;
                if(个体越界) 随机产生新的个体;
            endfor
            将遗传算法求取的测试方案数据保存到最后测试方案数据中,并更新测试数据集;
            if(有 IP 核的测试数据为空) 更新种群;
            遗传算法中某些中间变量清零;
        endwhile
    }
```

根据该遗传算法的运行过程可知,如果要想实现该遗传算法,必须重点解决以下几个问题:数据的存储结构、种群的编码与表示,及个体适应度的计算.

4.1 数据结构

遗传算法主要是实现将 IP 核输入输出数据相

同的位进行压缩,从而形成新的一组测试集.所以,算法运行过程中需要处理的数据量大,如果数据结构不合理,会影响遗传算法程序的设计与优化.而通过遗传算法求取的测试控制数据与测试数据同时具有数量大且复杂的特定,并且生成的数据决定着处理器如何进行 IP 核的测试与控制,因此,生成的新数据的存储结构也非常重要.根据遗传算法的需要,将原始测试数据按表 1 所示的数据结构保存;而根据测试的需要将遗传算法求取的数据按表 2 所示的数据结构保存.

表 1 原始测试数据

序号	IP 核名	输入位宽	输出位宽	测试集	校对集	功耗	时长
1	IP ₁	inW ₁	outW ₁	T ₁	C ₁	P ₁	L ₁
...	...	...	...	...	...	...	...
n	IP _n	inW _n	outW _n	T _n	C _n	P _n	L _n

表 2 遗传算法求取的测试数据

次序	测试 IP 核	输入控制	输出控制	测试集	校对集	时长	位宽
1	V ₁	iCT ₁	oCT ₁			L ₁	W ₁
2	V ₂	iCT ₂	oCT ₂	T	C	L ₂	W ₂
...	...	...	...			...	...

表 1 和表 2 中的时长即测试数据的个数,因为总线上所有 IP 核的测试时钟信号相同,可以用测试数据个数代替时长,从而评估测试时间;而功耗则是 IP 核的平均功耗.

根据表 2 的数据完成测试是按照其测试次序进行的.进行次序 1 测试时,先将测试 IP 核集 V₁ 内的 IP 核开启;再根据输入输出总线控制 iCT₁ 和 oCT₁ 的数据将所测试的 IP 核接入总线;最后开始输入测试数据,其测试数据为 T 数组中的前 L₁ × W₁ 位,而此次测试需要的总线位宽为 W₁.以下则是按照次序 1 的步骤依次进行测试,直到完成所有测试,再根据测试结果和校对集 C 进行故障诊断.

4.2 种群的编码与表示

由于每次测试的 IP 核不确定,其确定主要依据 IP 核的总功耗满足且总线需要在总线位宽范围内.因此,可以将所有 IP 核的不同排序作为种群的全部个体.如果测试 IP 核的个数为 n,则其不同的排序有 n! 种,将不同排序存储在二维数组中(每一行为一种排序),再将二维数组的每一行的首地址存储在指针数组变量 p 中.此时,则种群的每一个个体可

以用二进制数表示的 0 至 $n! - 1$ 表示,而个体染色体的长度 w (即二进制数位宽)满足 $n! \leq 2^w - 1$.

IP 核的排序的生成则通过将每一个 IP 核用一个连续且不同的数字指代,即所有的排序为不同数字的排序.此时,可以通过 for 循环生成,并保存在二维数组中.例如存在 3 个 IP 核时,用 1、2、3 分别代表不同 IP 核,排序存储在二维数组 $T[][3]$ 中,则实现代码为

```
for(a[1]=1,i=0;a[1]≤3;a[1]++)
for(a[2]=1;a[2]≤3;a[2]++)
for(a[3]=1;a[3]≤3;a[3]++)
{ if(check_different(a)
// check_different(a)为检测 a 数组内值全不同时返回 1
{ for(j=0;j≤3;j++) T[i][j]=a[j];
i++; } }
```

4.3 适应度函数

适应度函数不仅是用来评估个体的优劣的标准,而且其复杂度是遗传算法复杂度的主要组成部分.因此,为了能够正确评估个体的优劣,且使遗传算法的复杂尽可能小,本文用经过压缩后所减少的测试时间来表示个体的适应度.其计算公式为

$$F = \sum_{i=0}^{n-1} (x_i - 1) * L.$$

式中, L 表示此个体的测试时长; x_i 表示总线中第 i 位上接入的线数; n 表示此个体的总线位宽.个体的 $(x_0, x_1, \dots, x_{n-1})$ 与 L 的计算步骤如下所示.

- (1) 查找满足功耗约束的所有 IP 的十进制测试数据,看是否可以压缩,如果存在且测试长度足够,则进行压缩,并对测试数据进行处理;
- (2) 查找新的测试数据是否拥有测试数据位全为 1 或 0 的,如果存在则根据具体情况进行数据的合理压缩;
- (3) 查找是否存在哪些位上的 1 的个数相同,如果存在则通过将测试数据的次序经过合适调动后压缩;
- (4) 已经不能够再进行压缩的数据,则查看还有哪些 IP 核的测试数据未进行压缩,则考虑是否舍弃,对于比较长的 IP 核,在满足总线位宽的情况下保留;
- (5) 进行数据的整理,计算出所需的 $(x_0, x_1, \dots, x_{n-1})$ 与 L ,并且保存算法需要的数据.

5 遗传算法的仿真测试

该测试方法的核心在于能否通过遗传算法找到

最佳的测试集,因此,如果能够验证遗传算法具有查找最佳测试集的作用,也就能够说明该方法是可行的.而遗传算法的运行只需要所测试的 IP 核的相关数据,因此,可以按照前面所说的数据结构建立一个测试数据库,然后对测试数据库进行求解验证即可.

5.1 测试数据库的建立

为验证该算法的正确性,本文选取 3-8 译码器、8-3 编码器、全加器、减法器、乘法器、用户自定义 User_IP1 和 User_IP2 的输入数据进行验证,其测试数据如表 3 所示.且测试系统的总线位宽设为 16 位,而最大功耗为 12 mW.

表 3 新建测试数据集

标号	IP 核名	输入位宽	测试集	功耗/mW	时长
IP ₁	Decoder	3	T_1	2	8
IP ₂	Encoder	8	T_2	2	10
IP ₃	Adder	9	T_3	3	22
IP ₄	Subtractor	9	T_4	3	22
IP ₅	Multiplier	8	T_5	4	16
IP ₆	User_IP1	3	T_6	2	8
IP ₇	User_IP2	4	T_7	3	16

由表 3 所建立的测试数据集可知道各 IP 核测试所需要的各种数据.如通过表 3 中数据就可以很快的得知译码器的测试集保存在 3×8 的二维数组 T_1 中,因此需要经过 8 次测试数据输入才能够完成其测试,而 T_1 中的数据则是根据其真值表所得.

5.2 遗传算法运行求取的结果

经过遗传算法所得数据如表 4 所示,从表中可以很容易的看到测试总时长为 35 个时钟周期(每一次的控制转换也需要消耗一个时钟周期),而对测试总线的位宽需要仅 9 位.

表 4 遗传算法计算结果

测试次序	测试 IP 核集	输入总线控制	测试集	时长	总线位宽
1	IP(1,3,4,5)	inCT ₁	newT	8	6
2	IP(3,4,6,7)	inCT ₂		8	6
3	IP(2,3,4)	inCT ₃		6	9
4	IP(2,5)	inCT ₄		4	8
5	IP(5)	inCT ₅		4	4

对于所建立的测试数据,如果按照每一总线数据位接入一根测试数据线进行测试调度优化,则又可分为总线指定和总线分配两种不同方法,其测试

调度优化如图 4 所示. 其中图 4(a)为总线位宽指定为 16 位时的测试调度优化结果,从中可以看到所需测试总时间为 60 个时钟周期;而图 4(b)为总线位宽可以在 0 至 18 之间进行合理分配时的测试调度优化结果,其所需测试总时间为 48 个时钟周期.

通过比较几种不同方法的结果,不难看出基于遗传算法的多 IP 核并行测试方法不仅能够在测试时间上有效缩短,且能够压缩测试数据集和减少对总线位宽的需求.

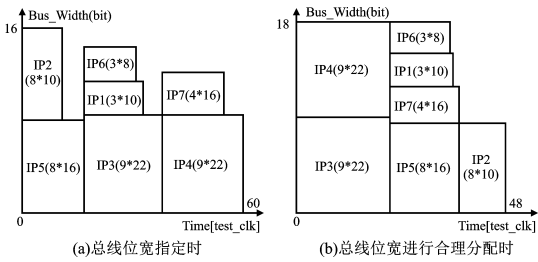


图 4 建立的测试数据的测试调度优化
(IP 核内的数字分别表示输入位宽和测试时长)

6 结束语

本文针对现有 SoC 测试方法的测试时间过长的问 题,提出了一种基于遗传算法的多 IP 核并行测试方法,为该方法的实现设计了相应的测试系统结构,并且介绍了测试系统的运行流程. 而对于测试流程,则重点介绍了遗传算法实现所存在的问题和解决方法. 最后通过应用遗传算法对所建立的测试数据库求取最佳测试方案的仿真,验证了该方法是有效可行的.

参考文献:

[1] 赵红军,阳日杰,崔坤林,等. 边界扫描技术的原理及其 应用[J]. 现代电子技术,2005,28(11):20-24.

[2] Giri CH, Tipparthi D K R, Chattopadhyay S. Genetic algorithm based approach for hierarchical SoC test scheduling[C]// Proceedings of the International Conference on Computing: Theory and Applications . Kolkata, 2007.

[3] 许川佩,胡红波. 基于量子粒子群的 SoC 测试调度优化[J]. 仪器仪表学报,2011,32(1):113-119.

[4] 雷加,方刚. 一种基于遗传算法的 SoC 测试调度方法 [J]. 仪器仪表学报,2007,28(4):15-43.

[5] 梁旭,李行善,于劲松. 基于遗传算法的并行测试调度 算法研究[J]. 电子测量与仪器学报, 2009, 23(2): 19-24.

[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): 226-231.

[7] 邓立宝,俞阳,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): 1238-1243.

[8] 雷加,苏波. 基于 IEEE1149. 4 标准 TAP 控制器的设计 [J]. 仪器仪表学报,2007,28(4):298-299.

作者简介:

谈恩民 男,(1966-),教授. 研究方向为计算机辅助测试和 集成电路可测性设计. E-mail: tem0135@guet. edu. cn.

贾亚平 男,(1990-),硕士研究生. 研究方向为集成电路可 测性设计.