

结合测试点质量的混合测试点集简约方法

欧阳彤^{1,2}, 许斌³, 董博文³, 周慧思¹, 张立明^{1,2}

(1. 吉林大学计算机科学与技术学院, 吉林长春 130012; 2. 符号计算与知识工程教育部重点实验室(吉林大学), 吉林长春 130012;
3. 吉林大学软件学院, 吉林长春 130012)

摘要: 集成电路中插入测试点是芯片测试中不可或缺的环节,其方法是通过在电路上插入一定数量的测试点来提高芯片的故障覆盖率. 集成电路测试是整个设计流程中不可或缺的关键步骤之一. 为了进一步缩短测试时间和提高芯片良品率,越来越多的国内外学者们从事集成电路测试方法的研究. 通过对插入测试点故障覆盖率的HTPI (Hybrid Test Point Insertion)方法深入研究,提出了结合测试点质量对测试点进行约简的RHTPI (Reduction of Hybrid Test Point Insertion)方法. 此方法在保证故障覆盖率相等的情况下,有效提高了测试点选择的效率. 从测试点覆盖率特征出发提出负质量测试点概念,依据负质量测试点对测试点集合进行约简,进而减小需要计算测试点集合的规模,有效缩短了求解时间. 为避免负质量测试点删除部分具有较高覆盖率的测试点,提出了结合自适应系数的自适应负质量测试点的概念,依据自适应负质量测试点对测试点集合进行约简,进而在保证故障覆盖率相等的情况下有效提高了测试点选择效率. 在标准测试用例上的实验结果表明,与HTPI方法相比,RHTPI方法计算候选测试点最小减少率为0.16%,最大减少率为44.56%,平均为24.56%,其求解效率最低提高了1.03,最高提高了2.37,平均提高了1.52,有效提高了测试点选择效率. 本文给出的方法有效减少了芯片测试时间,进而缩短了芯片设计周期.

关键词: 测试点插入;测试点;约简;电路测试;集成电路

基金项目: 国家自然科学基金(No.62076108, No.61872159, No.61672261);吉林省教育厅科学研究项目(No. JJKH20211106KJ, No. JJKH20211103KJ)

中图分类号: TP306

文献标识码: A

文章编号: 0372-2112(2023)06-1552-10

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20210710

Hybrid Test Point Set Reduction Method Based on Test Point Quality

OUYANG Dan-tong^{1,2}, XU Bin³, DONG Bo-wen³, ZHOU Hui-si¹, ZHANG Li-ming^{1,2}

(1. College of Computer Science and Technology, Jilin University, Changchun, Jilin 130012, China;

2. Key Laboratory of Symbolic Computation and Knowledge Engineering (Jilin University), Ministry of Education, Changchun, Jilin 130012, China;

3. Software Institute, Jilin University, Changchun, Jilin 130012, China)

Abstract: Inserting test points into integrated circuits is an indispensable link in chip testing, the method of which is to improve the fault coverage of the chip by inserting a certain number of test points on the circuit. Integrated circuit testing is one of the indispensable key steps in the whole design process. In order to further shorten the test time and improve the chip yield, more and more scholars at home and abroad are engaged in the research of integrated circuit test methods. Through the in-depth research on the htpi method of inserting test point fault coverage, a RHTPI (Reduction of Hybrid Test Point Insertion) method for reducing test points combined with test point quality is proposed. Under the condition of ensuring equal fault coverage, the efficiency of test point selection is effectively improved. Starting from the coverage characteristics of test points, the concept of negative quality test points is proposed, and the test point set is reduced according to the negative quality test points, so as to reduce the scale of the test point set to be calculated and effectively shorten the solution time. In order to avoid deleting some test points with high coverage from negative quality test points, the concept of adaptive negative quality test points combined with adaptive coefficient is proposed. The set of test points is reduced according to adaptive negative quality test points, and then the selection efficiency of test points is effectively improved under the condition of ensuring equal fault coverage. The experimental results on standard test case show that, compared with htpi method, the minimum reduction rate of candidate test points calculated by RHTPI method is 0.16%, the maximum reduction rate

is 44.56%, and the average is 24.56%; The minimum solution efficiency is increased by 1.03, the maximum is increased by 2.37, and the average is increased by 1.52, which effectively improves the efficiency of test point selection. The method given in this paper effectively reduces the chip test time, and then shortens the chip design cycle.

Key words: test point insertion; test point; reduction; circuit test; integrated circuit

Foundation Item(s): National Natural Science Foundation of China (No.62076108, No.61872159, No.61672261); Education Department of Jilin Province (No.JJKH20211106KJ, No.JJKH20211103KJ)

1 引言

随着现代集成电路工业的迅猛增长,集成电路规模增长到数十亿个晶体管级别.在电路的复杂度不断增加的情况下,其电路面积却在不断缩小,使电路故障测试面临更加严峻的挑战.为了缩短测试时间和提高芯片良品率,国内外学者们开始研究利用测试集和测试点的方法提高故障覆盖率.根据测试时是否改变电路的物理设计分类,测试方法主要包括基于测试向量的测试方法^[1-4]和在电路中物理插入测试点的测试方法.在工业界,以 Synopsys, Mentor Graphic 和 Cadence 三家公司为代表的芯片设计公司都提供商用的、基于测试向量方法的测试工具.在利用基于测试向量方法的测试后,电路中可能仍然存在某些故障,这些故障是不可测试的,此时需要从物理设计层面插入测试点提高故障覆盖率.

利用插入测试点方法进行测试时会对电路产生额外的开销(增加电路面积、功率和延迟),此方法先选择测试点的位置以及类型,尽量在插入测试点数目最小时最大化电路故障覆盖率.为了降低基于插入测试点的测试方法产生的额外的开销,国内外许多学者对测试点选择方法进行了研究. Hayes 等^[5]从减少故障检测测试次数的角度对测试点选择问题进行了研究,依据电路中是否有无扇形分叉的结构特征提出一种无扇出网络中选择最优测试点集的分枝定界算法,进而有效减少了故障检测次数. Iyengar 等^[6]提出一种结合伪随机模式的测试方法,先在逻辑综合中通过逻辑变换对电路进行简化,随后利用故障模拟器指导控制点和观测点的选择,对控制点和观测点使用共享和冷凝优化,有效减小了测试带来的面积开销. Eggersglüß 等^[7]提出结合可满足性求解器的测试点选择方法,利用可满足性求解器求解得到一组最小的有效测试点,此组测试点能检测到所有故障,对固定性故障具有 100% 的故障覆盖率. Brglez 等^[8]提出了一种可测试性算法,该算法为电路中的所有线路分配可控性和可观测性值,利用该值可以指导测试点的插入. Yoshimura 等^[9]基于改进的故障检测概率和赋值概率插入测试点,通过计算每条主输出线的可观测性成本,在具有低可观测性成本的线上插入测试点;实验结果表明,该方法在减少测试模式的数量上是非常有效的. Veena 等^[10]提出针对

较难检测故障的观测点选择方法对难以在主输出观测到的故障位置添加观测点,进而捕获较难观测的故障,获得更高的故障覆盖率. He 等^[11]提出了结合时间感知框架评估测试点的选择方法,根据测试点的效率对测试点进行排序并获得最有效测试点,进而缩短测试时间、优化区域开销以及提升测试故障覆盖率. Acero 和 Rajski 等^[12,13]提出基于识别和解决内部信号之间冲突的测试点方法,在最大内部冲突相对应的位置插入冲突感知测试点增加单个测试模式针对的故障数量,从而显著减少测试模式计数和生成的测试数据量. Chen 等^[14]结合可控制性和可观测性的测试点选择方法,考虑了信号在遇到重收敛的分支输出时的相互依赖性,以更准确地选择控制点.如果控制点可以增加电路的可测试性,则该控制点将被添加到线路中. Yang 等^[15]提出结合触发器的测试方法,在控制点附近的电路中利用已有触发器作为驱动器,将触发器作为测试点驱动器来减少插入控制点所带来的面积开销. Roy 等^[16,17]提出了与现有控制点和观测点不同的反演点测试方法,反演点使用异或门实现,当激活时可以同时激发和传播固定型故障,反演点比控制点和观测点更能增加故障覆盖率.

本文在对 Roy 等给出的 HTPI (Hybrid Test Point Insertion) 方法^[16]进行充分研究的基础上,发现了其在测试点插入方法上存在计算复杂性较高的不足:在每次测试点插入过程中需要对所有测试点进行计算.随着电路规模的扩大,所有测试点规模增大,求解时间也随之增大.若能对测试点集合的求解空间进行优化,减少测试点插入过程的求解耗时,将提高算法求解效率.针对此问题,本文提出一种结合测试点质量约简候选所有测试点的方法——RHTPI (Reduction of Hybrid Test Point Insertion) 方法. RHTPI 方法将结合测试点的质量对候选所有测试点进行约简,避免了上述测试点插入方法在每次测试点插入过程中对所有候选测试点进行计算,进而提高了算法的整体求解效率.本文的主要贡献包括:

(1) 提出了负质量测试点概念,依据负质量测试点对所有测试点进行约简,减小了需要计算测试点集合的规模.

(2) 提出了自适应系数的自适应负质量测试点.依

据自适应负质量测试点对测试点集合进行约简,进而在保证故障覆盖率相等的情况下有效提高了测试点选择效率.

(3)实验表明以上两个方法不仅能减小所有测试点的规模,优化求解空间,缩短算法求解时间,而且能保证 RHTPI 方法选取测试点的故障覆盖率与 HTPI 方法基本相等.

2 测试点结构

本节将介绍测试点的相关概念,并给出实例进行详细描述.

测试点在激活时对电路进行修改(禁用时不改变电路的功能),以增加电路的故障检测概率.测试点类型按逻辑功能主要包括控制点和观测点.下面先介绍控制点逻辑功能.

控制点在激活时强制改变电路的逻辑值为0或1,这有利于在应用随机测试向量时检测到原来不能检测到的故障.控制点在实现时使用额外的测试信号,当测试信号启用时原电路将被赋值逻辑0或1;当测试信号被禁用时,电路的功能不变.如图1(a)和(b)所示,控制点使用与门或者或门强制改变电路的逻辑功能.当测试信号输入为1时,图1(a)中原电路的最终输出被控制为1;当测试信号输入为0时,则不启用控制点功能,最终输出是原电路输出.当测试信号输入为0时,图1(b)中原电路的最终输出被控制为0;当测试信号输入为1时,则不启用控制点功能,最终输出的是原电路的输出.

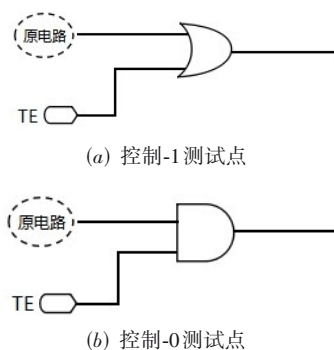


图1 与或门测试点电路图

第二类测试点是观测点.与控制点不同的是,观测点的逻辑功能是将观测点位置的逻辑值直接作为电路的输出被观测到.如图2所示,插入观测点的电路,观测点位置的逻辑值直接输出为电路的输出.

第三类测试点是反演测试点,是由 Roy 等^[16]在观测点和控制点基础上提出的.与控制点不同的是,反演点通过反演被控线路的值改变原电路信号的值.如图3

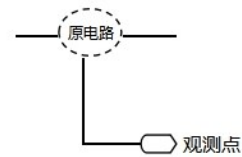


图2 观测测试点

所示,反演点使用异或门改变电路的逻辑功能.当原电路输出为0,测试信号为1时,原电路的最终输出被控制为1;当测试信号为0时,则不启用反演点功能,最终输出的是原电路的输出.当原电路输出为1,测试信号为1时,原电路的最终输出被控制为0;当测试信号为0时,则不启用反演点功能,最终输出的是原电路的输出.

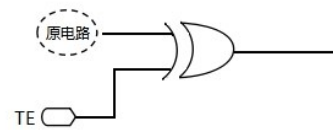


图3 反演测试点

在测试期间,启用控制点和观测点修改电路逻辑更易于在电路上实现,测试点插入方法被广泛地使用.表1显示了对C17电路施加一个测试向量时,无测试点以及启用观测点、控制点时的检测故障情况.从表中可以看出,启用观测点时,最低和最高检测故障个数分别为6和8,都不低于无测试点时的故障检测个数.但在启用控制点时,最低和最高检测故障个数分别为3和6,故障检测个数都不高于无测试点时的故障检测个数.这说明并不是所有的测试点都可以提高电路的故障检测率.因此如何选取有效的测试点成为提高故障检测覆盖率时值得研究的关键问题.

表1 C17电路检测故障数

电路	无测试点	观测点		控制点		故障总数
		最高	最低	最高	最低	
C17	6	8	6	6	3	22

3 HTPI方法

在合适的位置插入控制和观测测试点,能有效提高电路的故障覆盖率.为进一步提高故障覆盖率,Roy 等^[16]结合控制点和观测点的功能,提出了具有反演功能的测试点的 HTPI 方法,进而获得了比控制点和观测点更高的故障覆盖率.下面介绍 HTPI 方法的相关定义和算法,随后介绍此方法的测试点选择过程.

定义1^[16] 可控概率.对于电路中的某一线路,将在该线路上出现逻辑1的概率称为可控概率,记为CC;将在该线路上不出现逻辑1(出现逻辑0)的概率

记为 1-CC.

定义 2^[16] 可观测概率. 对于电路中的某一线路, 将观测到该线路的概率称为可观测概率, 记为 CO.

定义 3^[16] 故障检测概率. 对于电路中发生的故障为 f_k , 将 f_k 所在线路的可控概率与可观测概率的乘积称为故障检测概率, 记为 P_f .

依据上述定义, 对固定型 0 故障检测概率记为 $P_{f_0} = CC \times CO$, 对于固定型 1 故障检测概率记为 $P_{f_1} = (1 - CC) \times CO$. P_f 计算如下:

$$P_f = \begin{cases} CC \times CO, & \text{For SA0} \\ (1 - CC) \times CO, & \text{For SA1} \end{cases} \quad (1)$$

定义 4^[16] 故障覆盖率. 对于电路中故障 f_k, f_k 故障检测率 P_f 的累加和与所有故障个数 $|F|$ 的比值称为故障覆盖率, 即

$$TPCOV = \frac{\sum_{f \in F} P_f}{|F|} \quad (2)$$

HTPI 方法的主要思想是, 结合电路和测试点类型得到所有测试点, 计算所有测试点的故障覆盖率, 选取其中故障覆盖率最高的测试点加入候选测试点集合, 并将其从所有测试点中移除, 迭代求解直到达到测试点数目上限为止, 最后候选测试点集合中即为故障覆盖率最高的测试点集合. 下面介绍 HTPI 方法的伪代码 (算法 1).

算法 1 HTPI 算法

输入: TPLimit, Testpoints /*TPLimit 是测试点数目上限, Testpoints 是所有测试点集合*/

输出: Insert /*候选测试点集合*/

```

1. WHILE(Curnumber<TPLimit)
2.     FOR each testpoint∈testpoints
3.         TP←bestTP(testpoint); /*保存故障覆盖率最高的测试点*/
4.     END FOR
5.     Erase(TP); /*从候选测试点集中删除保存的测试点*/
6.     CTP←TP; /*将保存的测试点放入集合 CTP*/
7.     Active(TP); /*将保存的测试点激活*/
8.     Curnumber++;
9. END WHILE
10. FOR each testpoint∈CTP
11.     Deactive(testpoint);
12. END FOR
13. RETURN CTP

```

HTPI 方法主要过程如下: 当未达到测试点数目上限时 (第 1 行), 首先遍历所有测试点 (第 2~4 行), 计算它们的故障覆盖率, 保存其中故障覆盖率最高的测试点 (第 3 行), 放入候选测试点集合中 (第 6 行), 并将该

测试点从所有测试点中移除 (第 5 行), 进而保证 HTPI 方法不会重复选取同一个测试点放入候选测试点集合中; 随后将该测试点激活 (第 7 行), 迭代选择测试点故障覆盖率最高的测试点直到达到测试点数目上限; 最后对候选测试点集合中的测试点进行取消激活处理 (第 10~12 行), HTPI 方法返回候选测试点集合 (第 13 行).

当候选测试点激活时会对其他所有测试点的故障覆盖率产生影响, 因此需要对其他测试点的故障覆盖率进行重新计算. HTPI 方法在每次测试点插入过程中需要对其他所有测试点进行计算, 导致计算复杂度较高. 下面介绍本文提出的, 结合测试点质量约简测试点集合的 RHTPI 方法.

4 RHTPI 方法

本文针对 HTPI 方法每次对所有非候选测试点进行计算的这一不足, 提出了结合测试点质量进行约简的 RHTPI 方法. 方法基本思想如下: RHTPI 方法是选择故障覆盖率较高的测试点的方法, RHTPI 方法将 HTPI 方法中故障覆盖率较低的测试点进行删除, 进而避免对故障覆盖率较低的测试点进行计算.

下面介绍 RHTPI 方法相关定义和算法, 最后结合电路实例详细介绍 HTPI 方法和 RHTPI 方法的测试点求解过程.

4.1 结合负质量测试点的约简方法

定义 5 候选测试点. 在测试点选择过程中, 将从所有测试点中选择后插入电路的测试点集合称为候选测试点.

定义 6 故障覆盖基准值. 当电路中无候选测试点被激活时, 称此时的故障覆盖率为故障覆盖基准值, 记为 base.

定义 7 测试点质量. 对于电路中的某个测试点 tp_k , 将候选测试点激活后的故障覆盖率与故障覆盖基准值的差值称为测试点质量, 记为 $quality = TPCOV_{tp_k} - base$.

定义 8 负质量测试点. 在候选测试点集合中, 将测试点质量小于 0 的测试点称为负质量测试点, 记为 $worstp = \{tp_k | quality_{tp_k} < 0\}$.

下面给出结合测试点质量约简的实例. 如表 2 所示, 对 C17 电路分别采用 HTPI 与结合负质量测试点方法, 测试点数目上限为 2, 测试点种类为控制点、观测点以及反演点. O_11_19 以及 O_3_10 分别为选取的候选测试点. 其中“O”为测试点的种类, 表示该点为观测点; “_3_10”表示该测试点在 C17 中的位置.

HTPI 方法对所有测试点的故障覆盖率进行计算, 总的测试点计算数目为 135; 结合负质量测试点方法对

负质量测试点进行约简,约简后的测试点计算数目为 106. 在候选测试点的故障覆盖率相同的情况下,与 HTPI 方法相比,RHTPI 方法总的测试点计算数目减少

了 27.3%. 因此从所有测试点中约简负质量测试点,可以缩小所有测试点集合的规模,避免对负质量测试点重复计算,提高求解效率.

表 2 C17 电路候选测试点

测试点选择过程	HTPI 方法			RHTPI 方法		
	候选测试点	故障覆盖率/%	测试点计算数目	候选测试点	故障覆盖率/%	测试点计算数目
第一轮	O_11_19	28.444 6	68	O_11_19	28.444 6	68
第二轮	O_3_10	33.132 1	67	O_3_10	33.132 1	38

4.2 自适应质量测试点

结合负质量测试点能有效约简测试点求解空间,但在不同规模或功能的电路中,利用负质量测试点去约简时,存在将 HTPI 方法作为候选测试点的负质量测试点删除的情况. 出现此情况的原因在于激活一些测试点后,这些负质量测试点的故障覆盖率有了显著提高.

针对此问题进一步提出结合自适应质量测试点的约简方法,在保证较高故障覆盖率的情况下减少计算测试点的个数. 下面给出自适应质量测试点方法的相关概念和实例.

定义 9 可能选取点. 在候选测试点集合中,将测试点故障覆盖率在电路基准值及 $0.99 \times \text{base}$ 之间的测试点称为可能选取点,记为

$$\text{possibletp} = \{tp_k | 0.99 \times \text{base} < \text{TPCOV}_{tp_k} < \text{base}\}.$$

定义 10 在 RHTPI 方法中,将可能选取点的集合记为 possibletp , $\text{possibletp} = \{tp_k | tp_k \in \text{possibletp}\}$,将 possibletp 中 tp_k 的个数记为 $|\text{possibletp}|$.

定义 11 选取点比例. 将可能选取点个数 $|\text{possibletp}|$ 与负质量测试点个数 $|\text{worsetp}|$ 的比值称为选取点比例,记为 $P_{tp} = |\text{possibletp}| / |\text{worsetp}|$.

定义 12 自适应系数. 在 RHTPI 方法中,将自适应阈值记为 Limit ,将自适应系数记为 α .

当选取点比例 P_{tp} 大于自适应阈值 Limit 时,自适应系数 α 赋值为 0.99; 当选取点比例 P_{tp} 小于自适应阈值 Limit 时,自适应系数 α 赋值为 1, 即

$$\alpha = \begin{cases} 0.99, & \text{if } P_{tp} > \text{Limit} \\ 1, & \text{if } P_{tp} < \text{Limit} \end{cases} \quad (3)$$

定义 13 自适应质量. 将测试点激活后的故障覆盖率记为 TPCOV_{tp_k} , 自适应系数记为 α , 基准值为 base , 则称 $\text{Aquality} = \text{TPCOV}_{tp_k} - \alpha \times \text{base}$ 为自适应质量.

定义 14 自适应负质量测试点. 对于所有测试点,称自适应质量 Aquality 小于 0 的测试点为自适应负质量测试点. 对于电路中的任意测试点 tp_k , 将满足 $\text{Aworsetp} = \{tp_k | \text{Aquality}_{tp_k} < 0\}$ 称为自适应负质量测试点集合.

结合负质量测试点能有效约简测试点求解空间,但在不同规模或功能的电路中,利用负质量测试点去约简时,存在将 HTPI 方法作为候选测试点的负质量测试点删除的情况. 利用自适应阈值 Limit 能针对不同规模的电路,得到规模不同的选取点集合和比例不同的选择点. 自适应阈值的提出是为了根据自适应阈值和选取点比例来选择 HTPI 或 RHTPI 方法进行约简. 自适应系数如果过大则变成 HTPI 方法,如果过小则删除了较少的测试点,求解空间约简效果较差.

在不同规模电路测试点选择上,RHTPI 方法选取的候选测试点时,可能删除某些故障覆盖率较高的负质量测试点,导致故障覆盖率低于 HTPI 方法. 自适应质量测试点方法则可以保留被负质量约简方法误删除的测试点,保证了选取的候选测试点的故障覆盖率与 HTPI 方法基本一致.

下面以 C2670 电路为例,介绍分别采用结合负质量测试点与自适应质量测试点的约简方法. 如表 3 所示,测试点数目上限为 15,测试点种类为控制点、观测点以及反转点.

表 3 C2670 电路候选测试点故障覆盖率

结合自适应质量测试点方法		结合负质量测试点方法		覆盖率增长/%
候选测试点	故障覆盖率/%	候选测试点	故障覆盖率/%	
C1_1642_1897	11.648 6	O_3758_3806	11.485 6	1.4
C0_8	11.992 3	O_3048	11.761 8	2.0
O_3806	12.466 8	I_2113	12.035 0	3.6
O_1773_2033	15.483 9	O_2563	14.177 6	9.2

在结合负质量测试点的约简方法中,依据负质量测试点标准对所有测试点进行约简,在约简后的测试点集合上计算故障覆盖率. 此方法得到的 15 个测试点不包含负质量测试点,得到故障覆盖率最大值为 14.177 6%,最小值为 11.485 6%,约简的测试点数目包括 3 525 个控制点以及 1 799 个反转点.

在自适应质量测试点方法中,首先计算电路选取点比例,结合选取点比例和自适应阈值 Limit 得到自适应系数. 依据自适应系数对测试点进行约简,在约简后的测试点集合上计算故障覆盖率. 此方法得到的 15 个

测试点包含 2 个负质量测试点为 C1_1642_1897 以及 C1_1852_2135, 得到故障覆盖率最大值为 15.483 9%, 最小值为 11.648 6%, 约简的测试点数目包括 3 325 个控制点以及 1 776 个反演点. 与结合负质量测试点方法相比, 自适应质量测试点方法约简的测试点数目减少, 减少的控制点数目为 200, 减少的反演点数目为 23. 可以保留作为候选测试点的负质量测试点. 而结合负质量约简方法虽然约简的测试点数目增多, 但是约简了作为候选测试点的负质量测试点. 因此自适应质量测试点方法与结合负质量测试点方法相比, 虽然约简的测试点数目减少了 223, 但是能够保留作为候选测试点的负质量测试点, 从而提高候选测试点的故障覆盖率. 从表 3 可以得知, 与结合负质量测试点方法相比, 结合自适应质量测试点方法将故障覆盖率最少提高了 1.4%, 最多提高了 9.2%.

4.3 RHTPI 算法

RHTPI 方法的基本思想: 首先计算电路的选取点比例, 随后结合选取点比例和自适应阈值 Limit 得到自适应系数; 依据自适应系数对测试点进行约简, 在约简后的测试点集合上选择候选测试点. 下面给出该方法的伪代码(算法 2).

RHTPI 方法在第一轮候选测试点选取中(第 2 行), 首先计算电路选取点的比例(第 3 行), 当其大于自适应阈值 Limit 时, 自适应系数赋值为 0.99; 当其小于自适应阈值 Limit 时, 自适应系数赋值为 1(第 4~9 行). 结合自适应系数, 保存所有测试点中的自适应负质量测试点(第 10 行), 计算所有测试点的故障覆盖率, 保存故障覆盖率最高的测试点作为候选测试点(第 11 行), 从所有测试点中移除自适应负质量测试点(第 12 行). 将候选测试点从所有测试点中移除(第 17 行), 进而保证 RHTPI 方法不会重复选取同一个测试点放入候选测试点集合中. 随后将候选测试点放入集合 CTP 并将之激活(第 18 行). 在第二轮及以后的候选测试点选取中, 计算约简后所有测试点中故障覆盖率最高的测试点作为候选测试点, 直到达到测试点数目上限(第 14~16 行). 最后对候选测试点集合中的测试点进行取消激活处理(第 22 行), RHTPI 方法返回候选测试点集合(第 23 行).

下面以 C17 电路为例详细说明算法流程. 如图 4 所示, C17 电路具有 5 个输入 $\{P_1, P_3, P_6, P_2, P_7\}$ 以及 2 个输出 $\{O_1, O_2\}$, 测试点数目上限为 2, 测试点种类为控制点、观测点和反演点. 所有测试点集合包括 34 个控制点、17 个观测点以及 17 个反演点. 因为控制点分为控制-1 以及控制-0 两种类型, 所以它的数目为观测点或反演点的 2 倍.

RHTPI 方法首先计算电路选取点的比例为 0, 小于自适应阈值, 将自适应系数赋值为 1, 随后结合自适应

算法 2 RHTPI 算法

输入: TPLimit, Testpoints, Limit /*Limit 是自适应阈值*/

输出: CTP /*候选测试点集合*/

```

1. WHILE(Curnumber<TPLimit)
2.   IF(Curnumber= =0) /*第一次测试点选取过程*/
3.     pTP←calworseTP→possibleTP(testpoints); /*计算选取点的比例*/
4.     IF(pTP>Limit)
5.       α=0.99;
6.     END IF
7.     IF(pTP<limit)
8.       α=1;
9.     END IF
10.    worseTP←calworseTP→wTP(testpoints); /*保存负质量测试点*/
11.    TP←calworseTP→bestTP(testpoints); /*保存故障覆盖率最高的测试点*/
12.    Erase(worseTP); /*从所有测试点中移除自适应负质量测试点*/
13.  END IF
14.  IF(Curnumber>0)
15.    TP←bestTP(testpoints); /*保存故障覆盖率最高的测试点作为候选测试点*/
16.  END IF
17.  Erase(TP); /*从所有测试点中删除候选测试点*/
18.  CTP←TP; /*将候选测试点放入集合 CTP*/
19.  Active(TP); /*将候选测试点激活*/
20.  Curnumber++;
21. END WHILE
22. Deactive(TP);
23. RETURN CTP

```

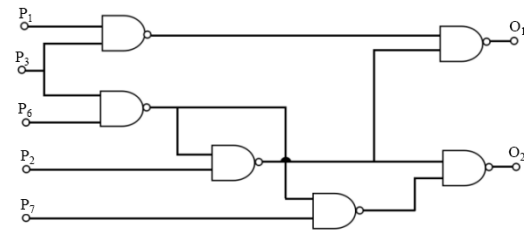


图 4 C17 电路图

系数得到自适应负质量测试点. 在第一次迭代计算时, 从 68 个所有测试点中保存 24 个控制点以及 5 个反演点作为自适应负质量测试点; 自适应负质量测试点包括从 C0_16(故障覆盖率为 6.250 0%)到 C1_2(故障覆盖率为 22.581 9%)在内的 24 个控制点以及从 I_10(故障覆盖率为 19.291 4%)到 I_16_23(故障覆盖率为 21.185 2%)在内的 5 个反演点. 在计算 68 个测试点后得到故障覆盖率最高的测试点为 O_11_19, 将其作为候

测试点. 在第二次迭代计算时,从约简后的 38 个测试点中计算故障覆盖率最高的测试点为 O_3_10,将其作为候选测试点;在后续的迭代计算过程中,在约简后(38 个)未被选择的测试点中继续选择故障覆盖率最高的测试点,直到达到测试点数目时返回候选测试点集合.

5 实验结果

在本节将提出的 RHTPI 方法与 Roy 等^[16]的 HTPI 方法进行比较,实验运行平台如下: DESKTOP-A19ALO6, Windows10, MSVC++ 编译器, Inter i7-4700 Processors, 2.60 GHz, 16 GB RAM. 测试用例选用文献[16]中的标准测试用例 ISCAS85 以及 ITC99. 在实验中,分别利用 HTPI 方法和 RHTPI 方法选择候选测试点,通过比较其运行时间和候选测试点平均故障覆盖率来测试 RHTPI 方法的有效性.

表 4 给出了 HTPI 方法和 RHTPI 方法在利用控制点和观察点选择候选测试点时的详细信息. 表中第 1 和 2 列分别为电路名称和电路中包含的元件个数,第 3 列为选择的候选测试点类型,第 4 和 5 列分别是两种方法的运行时间,第 6 列和第 7 列是两种方法选择的候选测试点的平均故障覆盖率. 第 8 和第 9 列是两种方法计算的测试点数目. 从表 4 可以看出, RHTPI 方法在 C6288, B05, B07, B11 中的平均故障覆盖率低于 HTPI 方法,这是因为 RHTPI 方法约简的自适应负质量测试点中包含 HTPI 的候选测试点,因此 RHTPI 方法选取的候选测试点的平均故障覆盖率略低于 HTPI 方法. 除上述这些电路外, HTPI 方法与 RHTPI 方法选取候选测试点的平均故障覆盖率相等. 这说明 RHTPI 方法从所有测试点中移除了冗余测试点,这些测试点在 HTPI 方法也未被选取为候选测试点,保证了其有效性.

表 4 两种测试点类型求解时间对比

Circuit details		Inserted TPs	Running Time/min		Average Fault Coverage/%		Number of TP Calculated	
Bench	Gates	C, O	HTPI	RHTPI	HTPI	RHTPI	HTPI	RHTPI
C432	203	2	1.63	1.15	10.85	10.85	2 591	2003
C499	275	2	1.26	0.95	28.58	28.58	2 993	2 398
C880	469	4	5.38	3.37	7.76	7.76	10 557	7 449
C1355	619	6	34.62	16.55	12.72	12.72	24 385	14 335
C1908	938	9	100.85	42.51	12.30	12.30	51 508	29 604
C2670	1 566	15	207.87	198.56	13.43	13.43	120 136	117 308
C3540	1 741	17	648.12	349.52	8.81	8.81	180 524	100 188
C5315	2 608	26	497.25	470.61	11.15	11.15	414 570	409 695
C6288	2 480	7	1 224.20	1 139.75	0.31	0.29	132 042	131 808
C7552	3 827	13	884.58	758.28	8.73	8.73	294 516	291 110
B03	190	2	0.20	0.13	21.07	21.07	3 983	3 516
B04	803	8	17.83	8.28	15.17	15.17	36 665	21 923
B05	1 032	10	51.48	38.86	11.56	11.08	67 761	47 335
B07	490	4	4.15	2.33	18.28	18.13	11 397	7 242
B08	204	2	0.38	0.26	16.81	16.81	2 351	1 829
B09	198	2	0.32	0.21	20.17	20.17	2 117	1 659
B10	223	2	0.40	0.28	15.02	15.02	2 705	2066
B11	801	8	20.82	8.91	10.41	10.26	38 945	22 187
B12	1 197	11	27.07	13.51	17.29	17.29	81 797	79 127
B13	415	4	0.68	0.43	20.54	20.54	8 769	5 634
B14	10 343	2	608.55	455.08	5.25	5.25	129 749	106 328
B15	9 371	3	1 183.95	743.70	8.65	8.65	181 044	127 294

图 5 更直观地显示了 RHTPI 方法与 HTPI 方法求解时间加速比以及计算测试点数目的减少率. 从图 5 可以看出, RHTPI 方法在 C1908 的时间提升最高(加速比为 2.37),这是因为 RHTPI 方法与 HTPI 方法相比,从所有测试点中移除的自适应负质量测试点较多,计算的测试点数目较少,因此求解效率提高较大. 在 B12 上提

升最小(加速比为 1.03),这源于 B12 电路选取点比例达到了 RHTPI 方法的阈值 Limit,因此使用 RHTPI 方法约简的自适应负质量测试点较少,求解效率提高不明显. 加速比最小为 1.03,最大为 2.37,平均加速比为 1.52. 计算测试点数目的减少率最小为 0.16%,最大为 44.56%,平均减少率为 24.56%.

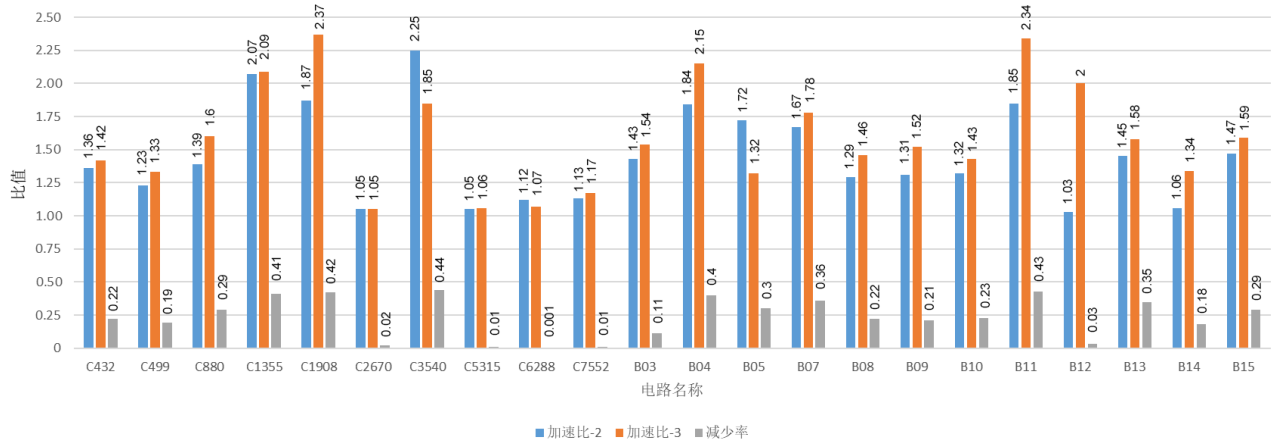


图5 求解时间加速比及计算测试点数目减少率

表5给出了HTPI方法和RHTPI方法在利用控制点、观察点和反演点选择候选测试点时的详细信息. 1~9例描述信息与表4相应的列描述信息相同. 从表5可以看出, RHTPI方法在C6288, B07, B11中的平均故障覆盖率低于HTPI方法.

图6中给出了选取点比例图. 在RHTPI方法中, 首

先对测试用例中的每个电路计算选择点比例; 随后与自适应阈值Limit进行比较, 依据式(3)计算自适应系数, 进而得到自适应负质量测试点. 在测试用例中选取点比例大于自适应阈值Limit的电路有C2670, C7552, C5315, B14, B15以及B12的控制点, 其他电路都小于自适应阈值Limit.

表5 三种测试点类型求解时间对比

Circuit details		Inserted TPs C, O, I	Running Time/min		Average Fault Coverage/%		Number of TP Calculated	
Bench	Gates		HTPI	RHTPI	HTPI	RHTPI	HTPI	RHTPI
C432	203	2	2.35	1.73	10.85	10.85	3 455	2 655
C499	275	2	1.90	1.55	28.58	28.58	3 991	3 378
C880	469	4	7.90	5.68	7.76	7.76	14 077	10 591
C1355	619	6	54.86	26.50	12.72	12.72	32 515	18 575
C1908	938	9	143.73	76.83	12.30	12.30	68 680	41 552
C2670	1 566	15	315.27	300.52	13.43	13.43	51 508	29 604
C3540	1 741	17	1 236.08	550.00	8.81	8.81	240 704	133 440
C5315	2 608	26	1 482.78	1 400.83	11.16	11.16	552 744	546 844
C6288	2 480	7	1 978.73	1 774.25	0.31	0.29	176 058	175 770
C7552	3 827	13	1 506.96	1 328.85	8.73	8.73	392 698	389 218
B03	190	2	0.30	0.21	21.07	21.07	4 647	4 057
B04	803	8	42.05	22.80	15.19	15.19	48 889	30 052
B05	1 032	10	120.35	69.80	11.04	11.04	90 351	54 783
B07	490	4	10.25	6.15	18.28	18.12	15 197	9 602
B08	204	2	0.58	0.45	16.81	16.81	3 135	2 485
B09	198	2	0.47	0.36	20.17	20.17	2 823	2 286
B10	223	2	1.03	0.78	15.02	15.02	3 607	2 815
B11	801	8	52.95	28.55	10.41	10.24	52 249	30 675
B12	1 197	11	71.08	68.83	17.38	17.38	109 066	94 736
B13	415	4	1.03	0.71	20.54	20.54	11 693	7 853
B14	10 343	2	1 530.97	1 449.51	5.25	5.25	172 999	141 175
B15	9 371	3	2 803.83	1 901.25	8.65	8.65	241 390	167 850

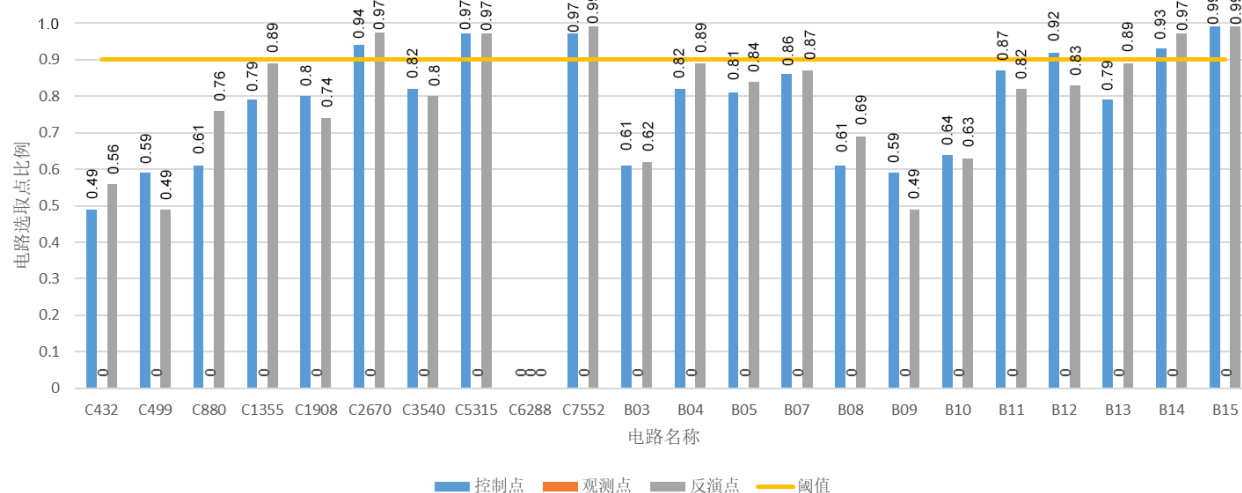


图6 选取点比例图

6 结束语

在电路的复杂度不断增加的情况下,其电路面积却在不断缩小,使电路故障测试的难度面临更加严峻的挑战.为了缩短测试时间和提高芯片良品率,国内外学者们对基于测试点的故障检测方法进行了研究.本文在对HTPI方法的研究基础上,提出了结合测试点质量对候选测试点集约简的测试点插入算法.首先从测试点覆盖率特征出发,提出了结合测试点质量对求解空间进行约简的RHTPI方法.为避免结合测试点质量进行约简时删除部分具有较高覆盖率的测试点,提出结合自适应系数的自适应负质量测试点的概念,依据自适应负质量测试点对测试点集合进行约简,进而有效缩减测试点集合的规模.实验结果表明,RHTPI方法测试点求解空间最小减少0.16%,最大减少44.56%,平均减少24.56%.在有效减小问题求解规模后,其求解效率最低提高了1.03倍,最高提高了2.37倍,平均提高了1.55倍,有效缩短了芯片测试时间,进而缩短了芯片设计周期.

参考文献

- [1] 欧阳丹彤,陈晓艳,叶靖,等.基于极小碰集求解算法的测试向量集约简[J].计算机研究与发展,2019,56(11):2448-2457.
OUYANG D T, CHEN X Y, YE J, et al. Test pattern set reduction based on the method of computing minimal hitting set[J]. Journal of Computer Research and Development, 2019, 56(11): 2448-2457. (in Chinese)
- [2] 詹文法,邵志伟.一种集成电路测试流程分级动态调整方法[J].电子学报,2020,48(8):1623-1630.
ZHAN W F, SHAO Z W. Hierarchical dynamic adjustment method for integrated circuit testing process[J]. Acta Electronica Sinica, 2020, 48(8): 1623-1630. (in Chinese)
- [3] 倪天明,常郝,卞景昌,等.基于边沿延时翻转的绑定前硅通孔测试方法[J].电子学报,2019,47(11):2278-2283.
NI T M, CHANG H, BIAN J C, et al. An edge transition delay based pre-bond TSV testing method[J]. Acta Electronica Sinica, 2019, 47(11): 2278-2283. (in Chinese)
- [4] NAG P K, GATTIKER A, WEI S C, et al. Modeling the economics of testing: A DFT perspective[J]. IEEE Design & Test of Computers, 2002, 19(1): 29-41.
- [5] HAYES J P, FRIEDMAN A D. Test point placement to simplify fault detection[J]. IEEE Transactions on Computers, 1974, C-23(7): 727-735.
- [6] IYENGAR V S, BRAND D. Synthesis of pseudo-random pattern testable designs[C]//Proceedings of 'Meeting the Tests of Time', International Test Conference. Washington: IEEE, 1989: 501-508.
- [7] EGGERSGLÜB S. Towards complete fault coverage by test point insertion using optimization-SAT techniques[C]//2019 IEEE International Test Conference in Asia. Tokyo: IEEE, 2019: 67-72.
- [8] BRGLEZ F. On testability analysis of combinational networks[C]//Proceedings of the IEEE International Symposium on Circuits and Systems. New York: IEEE, 1984: 221-225.
- [9] YOSHIMURA M, HOSOKAWA T, OHTA M. A test point insertion method to reduce the number of test patterns [C]//Proceedings of the 11th Asian Test Symposium. Guam: IEEE, 2002: 298-304.
- [10] VEENA V, PRABHU E, MOHAN N. Improved test coverage by observation point insertion for fault coverage analysis[C]//2019 3rd International Conference on Trends

in Electronics and Informatics (ICOEI). Tirunelveli: IEEE, 2019: 174-178.

- [11] HE M, CONTRERAS G K, TRAN D, et al. Test-point insertion efficiency analysis for LBIST in high-assurance applications[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2017, 25(9): 2602-2615.
- [12] ACERO C, FELTHAM D, HAPKE F, et al. Embedded deterministic test points for compact cell-aware tests[C]//2015 IEEE International Test Conference. Anaheim: IEEE, 2015: 1-8.
- [13] RAJSKI J, TYSZER J, ZAWADA J. On new class of test points and their applications[C]//2018 IEEE International Test Conference. Phoenix: IEEE, 2019: 1-9.
- [14] CHEN M J, XIANG D. Pseudorandom scan BIST using improved test point insertion techniques[C]//Proceedings of 7th International Conference on Solid-State and Integrated Circuits Technology. Beijing: IEEE, 2004: 2043-2046.
- [15] YANG J S, TOUBA N A, NADEAU-DOSTIE B. Test point insertion with control points driven by existing functional flip-flops[J]. IEEE Transactions on Computers, 2012, 61(10): 1473-1483.
- [16] ROY S, STIENE B, MILLICAN S K, et al. Improved pseudo-random fault coverage through inversions: A study on test point architectures[J]. Journal of Electronic Testing, 2020, 36(1): 123-133.
- [17] ROY S, STIENE B, MILLICAN S K, et al. Improved random pattern delay fault coverage using inversion test points[C]//2019 IEEE 28th North Atlantic Test Workshop. Burlington: IEEE, 2019: 206-211.



张立明(通讯作者) 男,1980年生于吉林省榆树市. 吉林大学博士. 主要研究方向为基于模型诊断、可满足性问题和测试诊断.

E-mail: limingzhang@jlu.edu.cn

作者简介



欧阳丹彤 女,1968年出生于吉林省长春市. 1998年毕业于吉林大学计算机科学系,并获博士学位. 博士,教授,博士生导师. 主要研究方向为自动推理与基于模型诊断.

E-mail: ouyd@jlu.edu.cn



许 斌 男,1997年出生于山东省枣庄市. 目前就读于吉林大学软件学院. 主要研究方向为基于模型诊断和测试诊断.

E-mail: binxu19@jlu.edu.cn