结合结构特征基于测试集重排序的故障诊断方法

欧阳丹彤[1,3],刘 扬[2],宋金彩[2],王浩然[4],张立明[1,3]
(1. 吉林大学计算机科学与技术学院,吉林长春 130012; 2. 吉林大学软件学院,吉林长春 130012; 3. 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; 4. 中国科学院大学计算机科学与技术学院,北京 100000)

摘 要: 故障诊断是集成电路领域中的重要研究方向,基于测试激励集方法求解候选故障诊断是目前较为高效的诊断方法,而GTreord是目前具有较高诊断准确性的方法. 在对GTreord方法深入研究的基础上,本文依据测试激励与候选故障诊断解之间的结构特征,通过分析电路故障输出响应,提出结合结构特征的测试激励集重排序的候选诊断(Reordering Test Default Diagnosis,RTDD)方法. 根据测试激励对生成候选故障诊断解集合的影响程度的不同,提出测试分数概念;通过比较电路的实际故障输出响应、无故障输出响应、模型故障输出响应,计算出测试激励的测试分数. 测试激励集依据测试分数进行重排序,并将重排序后的测试激励集用于故障诊断. 实验结果表明,与GTreord方法相比,RTDD方法提高了测试激励集重排序的效率,求解时间提高1~4个数量级;此外,在保障同样诊断准确性的情况下,RTDD方法有效减少了所需测试的激励个数.

关键词: 故障诊断; 测试分数; 测试激励集重排序
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112(2022)01-0063-09
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20200399

Fault Diagnosis Method Based on Test Set Reordering Combined with Structural Features

OUYANG Dan-tong[1,3],LIU Yang[2],SONG Jin-cai[2],WANG Hao-ran[4],ZHANG Li-ming[1,3]
(1. *College of Computer Science and Technology, Jilin University, Changchun, Jilin* 130012, *China*;
2. *Software Institute, Jilin University, Changchun, Jilin* 130012, *China*;
3. *Key Laboratory of Symbolic Computation and Knowledge Engineering(Jilin University), Ministry of Education, Changchun, Jilin* 130012, *China*;
4. *School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing* 100000, *China*)

Abstract: Fault diagnosis is a main direction in the research of integrated circuits which can solve candidate fault diagnosis based on test sets effectively. GTreord is a method with the best diagnostic accuracy currently. Based on deep analysis of the GTreord method, in this paper, a candidate diagnosis solution method based on test set reordering is proposed, referred as RTDD. RTDD method is based on the structural characteristics between the test and the candidate fault diagnosis solutions and analyzes the circuit fault output response. According to the different influence degrees of test on the generation of candidate fault diagnosis solution set, the notion of test score is presented. By comparing the actual fault output response, non-fault output response and model fault output response of the circuit, the test score of the test is obtained. The test set is reordered according to the test score, and the reordered test is applied to the fault diagnosis. Compared with the GTreord method, the experiments show that RTDD method improves the efficiency of test set reordering, and the running time is improved by 1-4 orders of magnitude. In addition, RTDD method effectively reduces the number of required test under the same diagnostic accuracy.

Key words: fault diagnosis; test score; test set reordering

1 引言

随着现代半导体工业的迅猛发展,如今集成电路已经发展到由数十亿个晶体管组成的规模. 电路的规模不断增大但电路尺寸却不断缩小,使得电路故障检

收稿日期:2020-04-27;修回日期:2021-01-04;责任编辑:孙瑶
基金项目:国家自然科学基金(No.62076108,No.61872159,No.61972360)

测与诊断的难度和成本也随之不断增加. 为了解决故障诊断效率相对低下的问题,学者们开始研究如何能够快速、有效地诊断出电路中发生的故障[1,2]. 根据电路的实际输出响应,学者们主要使用基于模型的诊断方法[3~5]和基于测试的诊断方法[6~8]对错误的输出响应进行分析,从而得到该集成电路的候选故障信息. 基于模型的诊断方法通过建立集成电路对应的模型,利用电路输入得到电路的预期行为,然后以预期行为与观测行为的差异作为诊断方法的输入,进而计算能够解释这种差异的候选诊断解. 基于测试的故障诊断方法则是利用测试激励集对单故障的实际输出响应与该电路的故障输出响应进行比较,匹配得到与故障输出响应最契合的故障,并将这些故障作为候选诊断解,组成候选故障集合.

近几年,许多学者对基于测试激励集的故障诊断方法进行了研究. Pomeranz[9]提出了SCOR方法和DD方法:SCOR方法通过比较电路在测试激励集下单故障与真实故障输出响应之间相同的端口个数,将该值作为该故障的评分,选取评分最高的故障组成候选故障诊断解;DD方法在SCOR方法的基础上通过对比同一测试激励集下同一输出端口的单故障与真实故障输出响应,将输出响应相同的故障添加到F_{same}集合中,再根据SCOR方法从F_{same}中选取评分最高的候选故障组成候选故障诊断解. Pomeranz[10]提出了ADD方法,通过调用DD方法得到一个初始候选故障集,逐个删除测试激励集中的每个测试激励,将每次删除测试激励后的测试激励集作为DD方法的新测试激励集,在上一次候选故障集基础上再次调用DD方法得到新的候选故障集,迭代求解直到得到最小的候选故障集. Pomeranz[11]随后提出了一种两阶段故障诊断方法:首先,使用完整的测试激励集作为故障诊断测试激励集调用DD方法得到一组候选故障集,将该候选故障集与故障诊断测试激励集作为ADD方法的输入得到一组新的候选故障集和新的故障诊断测试激励集,针对新的候选故障集中的无法区分故障对,使用诊断测试生成方法,产生可以区分故障对的新的测试激励,并将该测试激励添加到当前故障诊断测试激励集当中,形成新的故障诊断测试激励集;其次,将第一阶段得到的新故障诊断测试激励集与新候选故障集作为第二阶段DD方法的输入,之后的过程与第一阶段相同,将最后得到的候选故障集作为候选故障诊断解.

为进一步提高基于测试激励集的方法的故障诊断准确性,国内外学者通过分析测试激励与故障之间的关系,开展了关于通过调整测试激励集顺序来提高故障诊断准确性的研究. Bernardi等学者[12]提出通过对测试激励集进行重新排序的故障诊断方法,通过建立一个基于树的故障字典,从根节点到叶节点遍历诊断树来识别每个故障,测试重新排序用于最小化根节点到叶节点路径的长度,以减少故障诊断所需识别的故障信息. Bolchini等学者[13]通过数据挖掘技术提取测试与故障之间的规则,并结合已有的系统模型建立测试与故障之间的关系,根据该关系对测试激励集重排序,在保证故障诊断准确性的前提下较大程度减少了故障诊断过程所需的测试激励数目. Huang等学者[14]为了提高有限故障信息下的扫描链故障诊断分辨率,提出诊断覆盖率的概念,利用诊断覆盖率度量测试激励的诊断准确性,对测试激励按照诊断覆盖率大小进行降序重新排序,在保证诊断准确性的情况下有效缩减了测试激励集规模. Xue等学者[15]提出一种结合诊断特征的测试激励重排序方法,根据测试激励集对诊断结果的影响程度的大小对测试激励集进行排序,当遍历至对诊断结果无影响的测试激励时便停止诊断过程,得到候选故障集,将最终的候选故障集作为候选故障诊断解,进一步得到最终的诊断解. Bodhe等学者[16]提出了N-Cover方法,该方法使用贪婪算法选择故障数据,每次选择包含故障位最多的故障数据加入当前解中,直到达到特定的覆盖要求为止,尽量在满足覆盖要求的情况下有效缩减故障数据. Bodhe等学者[17]在N-Cover方法基础上提出了结合测试激励集重排序的GTreord方法,该方法在不影响诊断质量的情况下对测试激励重新排序,然后调用N-Cover方法,进而较大程度提高了故障诊断方法的准确性.

文献[9~11]提出的方法主要结合测试激励集与输出响应以有效提高故障诊断效率. 文献[12~17]中提出的方法则是利用基于测试激励集重排序的方法,提高故障诊断的准确率、分辨率以及诊断效率. 本文在对以GTreord为代表的基于测试激励集重排序的方法深入研究的基础上,依据每个测试激励对生成该故障诊断解集合的影响程度的不同,提出了测试分数概念,在此基础上提出基于测试激励集重排序的故障诊断方法(Reordering Test Default Diagnosis, RTDD). RTDD方法依据测试激励与候选故障诊断解之间的结构特征,得到每个测试激励的测试分数,根据测试分数对测试激励进行重新排序,并将重排序后的测试激励集用于故障诊断. RTDD方法相较于GTreord方法,主要有以下优点:

(1)RTDD方法有效缩短了测试激励集重排序时间,提高了测试激励集重排序效率;

(2)RTDD方法中的重排序测试激励集方法,在保证同样故障诊断准确性的情况下能有效减少测试激励个数.

2 GTreord方法

本节先介绍基于测试激励集重排序的故障诊断方法中用到的相关概念和定义,随后结合相关诊断实例详细介绍GTreord方法[17].

2.1 基本概念

下面介绍GTreord方法中所使用概念的基本定义.

定义1 电路中单模型故障为f_k,所有可能发生的模型故障集合为F,电路中真实发生的故障集合为f_{obs},用于诊断过程的单个测试激励为t_i,测试激励集合为T,记为$F=\{f_0,f_1,f_2,\cdots,f_{n-1},f_n\}$,$f_{\text{obs}}=\{f_k|f_k\in F\}$,$T=\{t_0,t_1,t_2,\cdots,t_{m-1},t_m\}$,其中$f_{\text{obs}}\subset F$.

定义2 电路在T下发生f_{obs}中故障时电路的实际输出响应为R_{obs},电路在T下无故障发生时的预期输出响应为R_{ff},电路在T下发生f_k时电路的实际输出响应为R_k.

定义3 电路的第j位(0≤j<输出端口总数)输出端口为z_j. 在电路在t_i激励下,发生f_{obs}中故障时电路z_j的实际输出响应为$R_{\text{obs}}(i,j)$,发生f_k时电路z_j的实际输出响应为$R_k(i,j)$,无故障发生时电路z_j的预期输出响应为$R_{\text{ff}}(i,j)$.

定义4 候选故障诊断解的集合为CND.

定义5 用于诊断f_k时保证CND中候选故障诊断解数量不变的最小测试子集为$T_{(\text{sub},k)}$,满足$T_{(\text{sub},k)}\subset T$. 称最小测试子集的集合为$T_{\text{SUB}}$,记为$T_{\text{SUB}}=\{T_{(\text{sub},0)},T_{(\text{sub},1)},T_{(\text{sub},2)},\cdots,T_{(\text{sub},n-1)},T_{(\text{sub},n)}\}$.

定义6 GTreord方法重排序测试激励集为T_{reord}. 使用T_{reord}作为N-Cover方法输入时,N-Cover方法返回的诊断测试激励集为T_{cover}.

2.2 GTreord方法

GTreord方法的两个过程是测试激励删除过程和测试激励集重排序过程. 测试激励删除过程通过删除t_i得到每个f_k对应的$T_{(\text{sub},k)}$,并由所有$T_{(\text{sub},k)}$组成T_{SUB};测试激励集重排序过程根据每个t_i在T_{SUB}中出现的次数,由高到低对T进行重排序. 排序后调用N-Cover方法得到T_{cover}. GTreord方法伪代码如算法1所示.

根据伪代码步骤2至步骤18为测试激励删除过程,该过程通过模拟每个f_k的发生,将f_k作为f_{obs}. 并将R_{obs},R_k,T,F作为DD方法[9]的输入进行故障诊断,得到初始CND. 之后从T的末尾开始逐个删除t_i,将删除t_i后的$T_{(\text{sub},k)}$代替T作为DD方法的输入得到CND_{new},若删除t_i后CND_{new}中的候选诊断解的数量不变则接受t_i的删除,否则将t_i重新加入T并将其置于T的首部,对T中t_i的下标进行更新. 遍历T后得到最终该f_k对应的$T_{(\text{sub},k)}$. 依照该过程遍历F后得到T_{SUB}. 下面结合例1对测试激励删除过程进行详细描述.

算法1 GTreord

输入:R_k, T, F

输出:T_{cover}

```
1.  Begin
2.    For all f_k ∈ F Do
3.      R_obs ← R_k; T_(sub,k) ← T; rem ← 0;
        i ← m; CND ← DD(R_obs, R_k, T, F);
4.      Remove t_i in T_(sub,k);
5.      CND_new ← DD(R_obs, R_k, T_(sub,k), F);
6.      If |CND_new| == |CND| Then
7.        i ← i − 1;
8.      Else
9.        将 t_i 置于 T_(sub,k) 的首部;
10.       更新 T_(sub,k) 中 t_i 的下标;
11.       rem ← rem + 1
12.     End If
13.     If i > rem Then
14.       Goto step 5;
15.     Else
16.       Add T_(sub,k) to T_SUB;
17.     End If
18.   End For
19.   For all t_i ∈ T Do
20.     s(t_i) ← 0;
21.     For all T_(sub,k) ∈ T_SUB Do
22.       If t_i ∈ T_(sub,k) Then
23.         s(t_i) ← s(t_i) + 1;
24.       End If
25.     End For
26.   End For
27.   T_reord ← bubble_sort(T) with s(t_i);
28.   T_cover ← N−Cover(T_reord, F, R_k);
29.   Return T_cover;
30.  End
```

例1 本实例中使用的电路为c17基准化电路. c17实例具体信息:电路中有2个可观测输出端口z_0和z_1,$F=\{f_0,f_1,f_2,\cdots,f_{12},f_{13}\}$,$T=\{t_0,t_1,t_2,\cdots,t_5,t_6\}$,具体信息如图1($a$)所示. 本实例中电路模拟$f_0$为真实故障时,使用$T$进行故障诊断得到的$\text{CND}=\{f_0,f_2,f_5,f_{11}\}$,$|\text{CND}|=4$.

下面详细介绍GTreord方法测试激励删除过程. 如图1所示,首先令$T_{\text{sub}}=T$,随后从图1(a)中删除t_6,得到图1(b)所示的T_{sub},使用该测试激励集进行故障诊断得到$\text{CND}_{\text{new}}=\{f_0,f_2,f_4,f_5,f_{11}\}$,与CND

相比，$\left|\mathrm{CND}_{\mathrm{new}}\right| \neq |\mathrm{CND}|$，所以将 t_6 加入到图 1(b)中所示的 T_{sub} 首部，如图 1(c)所示，并更新 T_{sub} 的顺序. 随后将图 1(c)中 t_6 删除，得到图 1(d)所示的 T_{sub}，用该测试激励集进行故障诊断，得到 $\mathrm{CND}_{\mathrm{new}}=\left\{f_0,f_1,f_3,f_{11}\right\}$，此时 $\left|\mathrm{CND}_{\mathrm{new}}\right| = |\mathrm{CND}|$，所以彻底删除 t_6，随后删除 t_5，得到图 1(e)中所示的 T_{sub}，用该测试激励集进行故障诊断，得到 $\mathrm{CND}_{\mathrm{new}}=\left\{f_0,f_4,f_9,f_{11},f_{13}\right\}$，此时 $\left|\mathrm{CND}_{\mathrm{new}}\right| \neq |\mathrm{CND}|$，所以将 t_5 加入到图 1(e)中所示的 T_{sub} 首部，如图 1(f)所示，并更新 T_{sub} 的顺序. 以此类推，得到针对该模型故障的 T_{sub}.

步骤 19 至步骤 29 为测试激励集重排序过程，该过程在测试激励删除过程模拟 F 中所有的 f_k，在得到 T_{sub} 的基础上，计算每个 t_i 在 T_{sub} 中出现的次数，依据出现次数的高低对 T 进行排序得到最终的 T_{reord}. 随后将 T_{reord}、F、R_k 作为 N-Cover 方法的输入，得到最终的 T_{cover}.

在例 1 的基础上，介绍 GTreord 方法的重排序过程，例 1 中得到的部分 $T_{(\mathrm{sub},k)}$ 如图 2 所示.

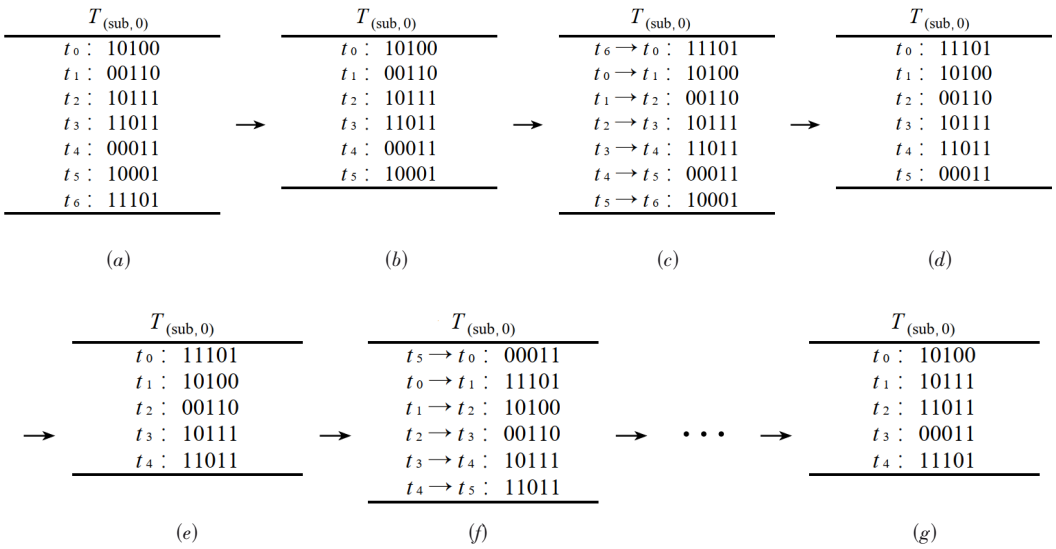

图 1 GTreord 方法测试激励删除过程

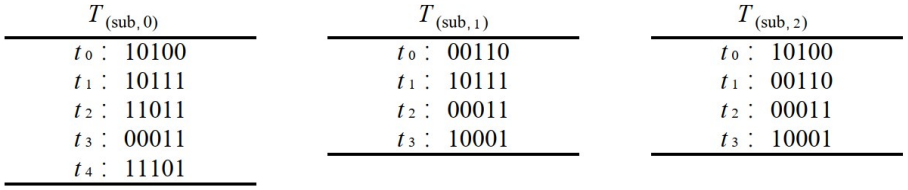

图 2 GTreord 方法部分 $T_{(\mathrm{sub},k)}$

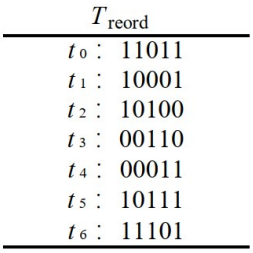

图 3 GTreord 方法测试激励集重排序结果

下面详细介绍 GTreord 方法测试激励集重排序过程. 以图 2 所示部分 T_{sub} 为例，T 中 t_0=10100，在 $T_{(\mathrm{sub},0)}$ 和 $T_{(\mathrm{sub},2)}$ 中均有出现，所以此时 t_0 的分数为 2. 以此类推，遍历 T_{sub} 得到所有 t_i 分数后进行排序，最终排序后的 T_{reord} 如图 3 所示. 随后将 T_{reord} 用于 N-Cover 方法得到 T_{cover}，$T_{\mathrm{cover}}=\left\{t_0,t_1,t_2,t_3,t_4\right\}$.

3 RTDD 方法

本节将首先介绍 RTDD 方法在测试激励集重排序过程中所使用概念的基本定义，然后结合伪代码与具体实例详细介绍 RTDD 方法.

3.1 基本概念

下面介绍 RTDD 方法相关定义.

定义 7 在 RTDD 方法中，将模型故障的集合记为

$F_{\text{new}}, F_{\text{new}}=\{f_K|f_K\in F\}$,且满足$F_{\text{new}}\subset F$. 将$F_{\text{new}}$中$f_K$的个数记为$|F_{\text{new}}|$.

定义 8 将F_{new}中在t_i的激励下满足$R_k\neq R_{\text{ff}}$的f_k组成的故障集合称为候选单故障集合$\text{CanF}(t_i)$. 将$\text{CanF}(t_i)$中f_k的个数记为$|\text{CanF}(t_i)|$. 且当对f_k进行诊断时,将候选单故障集合称为$\text{CanF}(t_i,k)$.

定义 9 在t_i的激励下,将$|\text{CanF}(t_i)|$与$|F_{\text{new}}|$的比值称为t_i的故障检测评分$\det(t_i)$,记为$\det(t_i)=|\text{CanF}(t_i)|/|F_{\text{new}}|$. 且当对$f_k$进行诊断时,将故障检测评分称为$\det(t_i,k)$,记为$\det(t_i,k)=|\text{CanF}(t_i,k)|/|F_{\text{new}}|$.

定义 10 在t_i的激励下,将F_{new}中所有f_k满足$R_k(i,j)=R_{\text{obs}}(i,j)$的所有元素个数称为$t_i$的故障诊断评分$\text{dia}(t_i)$,记为$\text{dia}(t_i)=|\{(i,j):R_k(i,j)=R_{\text{obs}}(i,j)\}|$. 且当对$f_k$进行诊断时,将故障诊断评分称为$\text{dia}(t_i,k)$,记为$\text{dia}(t_i,k)=|\{(i,j):R_k(i,j)=R_{\text{obs}}(i,j)\}|$.

定义 11 当对f_k进行诊断时,将t_i的故障检测评分与故障诊断评分的乘积称为测试分数$s(t_i,k)$,记为$s(t_i,k)=\det(t_i,k)\times\text{dia}(t_i,k)$,且将$F$中所有$f_k$对应的$s(t_i,k)$的和记为$s(t_i)$.

定义 12 将RTDD方法重排序测试激励集记为RT_{reord}. 将使用RT_{reord}作为N-Cover方法输入时返回的诊断测试激励集记为RT_{cover}.

3.2 RTDD方法

RTDD方法考虑到每个t_i对生成该故障诊断解集合的影响程度不同,提出了测试分数概念,依据t_i与候选故障诊断解之间的对应关系,通过比较在t_i的激励下电路的R_k、R_{obs}以及R_{ff},得到每个t_i对得到该故障诊断解集合的影响程度,即为测试分数,根据测试分数对t_i进行重新排序得到测试激励影响程度从高到低排列的新的测试激励集. 得到重排序测试激励集之后,将其用于N-Cover方法,得到N-Cover方法返回的诊断测试激励集.

在测试集重排序过程中,$s(t_i)$能够作为t_i的测试分数的原因有以下几点:首先,根据t_i与f_k之间的对应关系,一个t_i可以检测到多个f_k,一个f_k也可以被多个t_i检测,但一个t_i并不能检测到所有f_k. 且在t_i激励下满足$R_{\text{ff}}\neq R_k$的f_k更可能是故障诊断解,$\det(t_i)$表示t_i在CND中能够检测的f_k占候选故障诊断解的比重,即$\det(t_i)$可以表示该T的故障检测能力对生成该CND的影响程度;其次,在故障诊断方法中证明了若f_k的R_k与R_{obs}具有相同输出端口的个数越多,则该f_k更可能是故障诊断解,$\text{dia}(t_i)$为F_{new}中所有f_k在t_i激励下满足$R_k=R_{\text{obs}}$的端口个数之和,代表着该t_i在故障诊断过程中对生成CND的影响程度. 因此本文提出一个观点:若一个t_i能够检测到的f_k占候选故障诊断解比重越大且在该t_i激励下所有候选故障的R_k与R_{obs}相同输出端口个数之和越大,那么该t_i对生成该CND的影响程度越大. 所以本文结合t_i的$\det(t_i)$与$\text{dia}(t_i)$来计算t_i的测试分数.

RTDD方法的伪代码如算法2所示.

根据伪代码,RTDD方法对F中每一个f_k进行故障诊断. 首先,令$f_{\text{obs}}=f_k$,$R_{\text{obs}}=R_k$,使用T、F、R_{obs}作为故障诊断程序DD方法的输入,得到初始CND,将该CND作为测试激励集重排序过程的故障集合F_{new};其次,比较F_{new}中的f_k在T中的每一个t_i激励下,电路的R_k与R_{ff},将满足$R_{\text{ff}}\neq R_k$条件的f_k组成针对t_i激励下的$\text{CanF}(t_i,k)$;然后,计算$\det(t_i,k)$,即计算$\det(t_i,k)=|\text{CanF}(t_i,k)|/|F_{\text{new}}|$的值;接下来,计算$\text{dia}(t_i,k)$,即计算满足$t_i$的激励下,$F_{\text{new}}$中所有$f_k$满足$R_k(i,j)=R_{\text{obs}}(i,j)$条件的输出端口的个数之和;然后,计算$t_i$的$s(t_i,k)$,即计算$\det(t_i,k)$与$\text{dia}(t_i,k)$的乘积;下一步,遍历$F$中每一个$f_k$得到所有的$s(t_i,k)$,将同一$t_i$的$s(t_i,k)$相加,得到该$t_i$的$s(t_i)$;随后根据$s(t_i)$由高到低对测试激励进行重新排序,得到新的测试激励集RT_{reord};最后,将RT_{reord},F,R_k作为N-Cover方法的输入,得到满足N-Cover方法终止条件时所需t_i组成的RT_{cover}.

复杂性分析:设测试激励对电路进行故障检测的复杂度为$o(t)$,故障诊断和故障检测都是通过对比电路输出是否相同来实现,即故障诊断复杂度也为$o(t)$. 设故障个数为n,测试激励个数为m,GTreord方法中故障f_i对应的测试激励个数为d_i,$0<d_i<m$. 所有测试激励个数平均值记为$d=(d_0+d_1+\cdots+d_n)/n$,则测试激励总数为$n\times d$. GTreord方法需对每个故障检测每个测试激励是否可以删除,其求解复杂度为$n\times d\times o(t)$;其最坏情况下,不存在可以删除的测试激励时,其时间复杂度为$n\times(m-1)\times o(t)$. RTDD方法对m个测试激励求解故障检测和故障诊断分数,求解复杂度为$m\times(o(t)+o(t))$,即$2\times m\times o(t)$. 随着电路规模的增大,$n\times d$和$n\times(m-1)$的

算法2 RTDD

输入：R_k, R_{ff}, T, F

输出：$\mathrm{RT}_{\mathrm{reord}}$

```
Begin
  s(t_i) ← 0;
  For all f_k ∈ F Do
    R_obs ← R_k;
    F_new ← DD(R_obs, R_k, T, F);
    For all t_i ∈ T Do
      Samebit ← 0;
      For all f_k ∈ F_new Do
        If R_k ≠ R_ff Then
          Add f_k to CanF(t_i, k);
        End If
      End For
      det(t_i, k) ← |CanF(t_i, k)| / |F_new|;
      For all f_k ∈ F_new Do
        For all z_j Then
          If R_k(i, j) == R_obs(i, j) Then
            Samebit ← Samebit + 1;
          End If
        End For
      End For
      dia(t_i, k) ← Samebit;
      s(t_i, k) ← det(t_i, k) × dia(t_i, k);
    End For
    s(t_i) ← s(t_i) + s(t_i, k);
  End For
  RT_reord ← bubble_sort(T) with s(t_i);
  RT_cover ← N − Cover(RT_reord, F, R_k);
  Return RT_cover;
End
```

增加速度远高于$2\times m$，RTDD方法比GTreord方法的求解优势更明显.

下面结合例1中c17电路详细介绍RTDD方法具体过程. 电路模拟f_0为真实故障时，部分R_{obs}和R_k与T对应的具体细节如表1所示.

表1 c17部分R_k和R_{obs}对照表

t_i	R_{ff}	R_{obs}	R_0	R_1	R_2
t_0 = 10100	10	00	00	10	10
t_1 = 00110	00	00	00	10	00
t_2= 10111	00	00	00	10	10
t_3= 11011	11	11	11	11	01
t_4= 00011	01	01	01	01	01
t_5= 10001	01	01	01	01	01
t_6= 11101	11	11	11	11	11

下面详细介绍RTDD方法测试激励集重排序过程. 在例1中f_{obs}=f_0，经DD方法得到CND=$\{f_0,f_2,f_5,f_{11}\}$. 令F_{new}=CND，以表1中的信息为例，此时仅考虑故障集合F_{new}中的故障，以t_0的激励下计算过程为例，首先计算得到$\mathrm{CanF}(t_0,0)$，R_{ff}=10，R_0=00，R_2=10，其中R_{ff}=R_2，所以R_2不能添加到$\mathrm{CanF}(t_0,0)$中，以此类推遍历F_{new}中所有模型故障得到$\mathrm{CanF}(t_0,0)=\{f_0,f_{11}\}$；接下来计算$\det(t_i,0)$，根据已经得到的$\mathrm{CanF}(t_0,0)$与$F_{\mathrm{new}}$，$\det(t_0,0)=\dfrac{|\mathrm{CanF}(t_0,0)|}{|F_{\mathrm{new}}|}$=0.5；然后计算$\mathrm{dia}(t_0,0)$，在$t_0$的激励下$R_{\mathrm{obs}}$=00，$R_0$=00，$R_2$=10，由于$R_{\mathrm{obs}}(0,0)=R_0(0,0)$，$R_{\mathrm{obs}}(0,1)=R_0(0,1)$，$R_{\mathrm{obs}}(0,0)\neq R_2(0,0)$，$R_{\mathrm{obs}}(0,1)=R_2(0,1)$，此时$\mathrm{dia}(t_0,0)$=1+1+1=3，以此类推最终$\mathrm{dia}(t_0,0)$=5；下一步计算$s(t_0,0)$，根据计算得到的$t_0$对应的$\det(t_0,0)$和$\mathrm{dia}(t_0,0)$，计算$s(t_0,0)$=$\det(t_0,0)\times\mathrm{dia}(t_0,0)$=0.5×5=2.5；遍历$T$中所有的$t_i$，得到$f_{\mathrm{obs}}$=$f_0$时所有$t_i$的$s(t_i,0)$；遍历$F$中所有$f_k$，将同一$t_i$的$s(t_i,k)$相加，得到最终的$s(t_i)$；最后依据$s(t_i)$的大小进行测试激励集重排序，得到$\mathrm{RT}_{\mathrm{reord}}$；在得到$\mathrm{RT}_{\mathrm{reord}}$之后，将$\mathrm{RT}_{\mathrm{reord}}$，$F$，$R_k$作为N-Cover方法的输入，得到N-Cover方法返回的$\mathrm{RT}_{\mathrm{cover}}$. $\mathrm{RT}_{\mathrm{reord}}$与$\mathrm{RT}_{\mathrm{cover}}$如图4所示.

$\mathrm{RT}_{\mathrm{reord}}$
$t_0\rightarrow t_0$： 10100
$t_3\rightarrow t_1$： 11011
$t_4\rightarrow t_2$： 00011
$t_5\rightarrow t_3$： 10001
$t_1\rightarrow t_4$： 00110
$t_2\rightarrow t_5$： 10111
$t_6\rightarrow t_6$： 11101

$\mathrm{RT}_{\mathrm{cover}}$
t_0： 10100
t_1： 11011
t_2： 00011
t_3： 10001

图4 c17实例$\mathrm{RT}_{\mathrm{reord}}$，$\mathrm{RT}_{\mathrm{cover}}$

以上便是RTDD方法的具体实现过程. 从以上实例可知，本文提出的RTDD方法相较于GTreord方法能够更加高效地完成测试激励集重排序过程，且使用更

少的t_i即可达到与GTreord方法同样的诊断要求. 下一节将介绍RTDD方法和GTreord方法在ISCAS-85[18]和ISCAS-89[19]基准电路上的实验结果.

4　实验

本节将从测试激励集重排序效率和排序后测试激励集诊断准确性两个方面对RTDD方法的性能进行实验分析.

4.1　实验环境

实验环境如下:Windows 10, CPU Intel(R) Core(TM) i7-7700HQ 2.80GHz, 16GB RAM, python3. 使用商业工具Synopspy Tetramax2018生成T,使用商业工具Modelsim SE 10.4进行故障模拟仿真,得到电路模拟发生故障时的实际输出响应R_k与R_{obs}.

4.2　故障类型

本文所使用的故障是Stuck-at故障(固定型故障),使用该故障的原因有以下几点:

(1)多数其他类型故障都可以转换为Stuck-at故障或用不同Stuck-at故障的组合表示;

(2)一个故障诊断方法若对Stuck-at故障进行诊断得到较高的分辨率和准确率时,对于电路中的其他类型故障进行诊断也有较高的分辨率和准确率;

(3)其易于生成故障列表,且故障总数随着电路中的逻辑门的个数线性增长,故障列表的规模容易控制.

4.3　实验结果

本文采用的集成电路为国际测试领域中广泛使用的基准化集成电路ISCAS-85[18]和ISCAS-89[19]基准电路. 因为RTDD方法和GTreord方法的测试激励集重排序结果仅与测试激励集相关,不会受到其他因素的影响,所以仅进行一次对比实验. 下面对GTreord方法和本文提出的RTDD方法在完全一致的条件下的实验结果进行分析.

表2表示ISCAS-85基准下所有电路使用RTDD方法的测试激励集重排序过程与GTreord方法的结果对比. 其中第1列Circuit Name表示电路名称;第2列F_{num}表示该电路用于RTDD方法与GTreord方法的f_k的数量;第3列T_{num}表示电路初始T中t_i的个数;第4列和第5列分别表示RTDD方法和GTreord方法调用DD方法的次数;第6列R_{test}表示RTDD方法中RT_{cover}中t_i个数与RT_{reord}中t_i个数的比值;第7列G_{test}表示将GTreord方法中T_{cover}中t_i个数与T_{reord}中t_i个数的比值.

表3表示ISCAS-89基准下所有电路使用RTDD方法的测试激励集重排序过程与GTreord方法的结果对比,各列内容与表2相同.

从表2和表3中的RTDD方法测试激励集重排序过

表2　ISCAS-85实例测试激励集重排序结果

Circuit Name	F_{num}	T_{num}	DD_{time}		R_{test}	G_{test}
			RTDD	GTreord		
c17	14	7	14	112	57.1%	71.4%
c432	86	23	86	2064	65.2%	60.9%
c499	146	16	146	2482	75.0%	81.3%
c880	165	22	165	3795	72.7%	72.7%
c1355	146	15	146	2336	66.7%	66.7%
c1908	116	14	116	1740	50.0%	57.1%
c2670	589	48	589	28861	60.4%	62.5%
c3540	144	36	144	5328	63.9%	61.1%
c5315	596	38	596	23244	65.8%	68.4%
c6288	128	8	128	1152	75.0%	62.5%

表3　ISCAS-89实例测试激励集重排序结果

Circuit Name	F_{num}	T_{num}	DD_{time}		R_{test}	G_{test}
			RTDD	GTreord		
s27	32	10	32	352	70.0%	70.0%
s208_1	217	46	217	10199	69.6%	73.9%
s298	308	43	308	13552	58.1%	55.8%
s344	342	35	342	12312	48.6%	51.4%
s349	350	36	350	12950	77.8%	75.0%
s382	399	44	399	17955	68.2%	75.0%
s386	384	84	384	32640	72.6%	71.4%
s400	424	48	424	20776	62.5%	66.7%
s420	430	73	430	31820	56.2%	60.3%
s420_1	455	89	455	40950	56.2%	60.7%
s444	474	44	474	21330	65.9%	63.6%
s510	564	69	564	39480	71.0%	73.9%
s526	555	83	555	46620	75.9%	80.7%
s526n	553	78	553	43687	74.4%	70.5%
s641	463	84	463	39355	71.4%	72.6%
s713	581	84	581	49385	73.8%	75.0%
s820	850	138	850	118150	59.4%	61.6%
s832	870	142	870	124410	69.0%	68.3%
s838	857	128	857	110553	72.7%	74.2%
s838_1	931	186	931	174097	72.6%	80.1%
s953	1033	114	1033	118795	62.3%	61.4%
s1196	1240	189	1240	235600	73.0%	76.7%
s1238	1353	197	1353	267894	84.8%	84.8%
s1423	1515	107	1515	163620	67.3%	74.8%
s1488	1486	172	1486	257078	70.3%	75.6%
s1494	1506	168	1506	254514	68.5%	71.4%
s5378	4551	336	4551	1533687	65.2%	68.3%
s35932	39094	80	39094	3166614	74.5%	73.1%
s38417	30900	1050	30900	27737020	63.9%	66.0%
S38584	36299	763	36299	27732436	67.7%	68.4%

程与GTreord方法的实验结果分析如下.

(1)针对ISCAS-85基准电路和ISCAS-89基准电路,在达到同样诊断准确性情况下,比较RTDD方法的RT_{reord}与GTreord方法的T_{reord},虽然RT_{cover}中t_i的个数在电路c432,c3540,c6288,s298,s349,s386,s444,s526n,s832,s953,s35932电路上多于T_{cover}中t_i的个数,但是在ISCAS-85和ISCAS-89基准下,72.5%的电路上少于T_{cover}中t_i的个数.结果表明,针对大多数电路,RTDD方法测试激励集重排序过程得到的RT_{reord}与GTreord方法测试激励集重排序得到的T_{reord}相比,达到同样诊断准确性时需要用于诊断的测试激励个数更少,RTDD方法测试激励集重排序过程得到的RT_{reord}确实具有更好的诊断准确性.

(2)针对ISCAS-85基准电路和ISCAS-89基准电路,RTDD方法相较于GTreord方法调用DD方法的次数更少,节省了大量的时间成本.根据两种方法的算法流程可以得到以下推论:每个电路进行测试激励集重排序时,RTDD方法的测试激励集重排序过程需要调用DD方法的次数等于其F_{num};GTreord方法需要调用DD方法的次数等于其$F_{num}\times(T_{num}+1)$.尽管RTDD方法测试激励集重排序过程调用DD方法之后需要进行一系列的计算过程,但该过程所需时间与调用一次DD方法的时间相比而言是可以完全可以接受的.表4表示在ISCAS-85和ISCAS-89基本电路中,RTDD方法计算一次$s(t_i)$的过程所需时间,以及使用完整T调用一次DD方法的时间,其中Circuit Name表示电路名称;r_{time}表示RTDD方法计算过程的时间,d_{time}表示调用一次DD方法的时间,时间单位均为s.尽管GTreord方法在调用DD方法时的T不断减小,DD方法所需时间在不断减小,但是相较于RTDD方法的计算过程的时间仍然很长.总之,结果表明,RTDD方法测试激励集重排序过程具有更好的测试激励集重排序效率.

表4 RTDD方法测试激励集重排序过程时间对比

Circuit Name	r_{time}	d_{time}	Circuit name	r_{time}	d_{time}
c17	0.001	0.002	s444	0.083	21.524
c432	0.018	0.117	s510	0.092	18.364
c499	0.022	1.037	s526	0.172	59.765
c880	0.025	1.475	s526n	0.158	51.779
c1355	0.035	1.101	s641	0.090	72.297
c1908	0.014	0.407	s713	0.357	115.160
c2670	0.104	60.708	s820	0.615	228.971
c3540	0.047	1.321	s832	0.798	216.724
c5315	0.199	180.798	s838	0.143	306.586
c6288	0.015	0.548	s838_1	0.398	454.973
s27	0.002	0.051	s953	0.266	334.309
s208_1	0.024	1.246	s1196	0.462	819.920
s298	0.089	6.085	s1238	1.013	1023.137
s344	0.031	8.524	s1423	0.985	1769.468
s349	0.085	9.849	s1488	1.549	803.337
s382	0.116	15.286	s1494	1.429	857.503
s386	0.164	12.189	s5378	8.676	5320.265
s400	0.148	18.623	s35932	17.632	10658.890
s420	0.090	20.202	s38417	152.733	93450.989
s420_1	0.070	20.836	S38584	152.691	93455.521

综上所述,本文提出的RTDD方法与Gtreord方法相比有以下两点优势:

(1)RTDD方法能够更加高效地对测试激励集进行重排序,时间缩短最多可达到4个数量级;

(2)RTDD方法在保障同样诊断准确性的情况下,72.5%的电路所需t_i更少,平均减少3.645%,具有更强的诊断准确性.

5 总结

本文提出的RTDD方法根据测试激励与候选故障诊断解之间的关系计算测试激励的测试分数,依据测试分数对测试激励集进行重排序,并将用重排序后的测试激励集用故障诊断.在标准测试用例上的实验表明:RTDD方法的测试激励集重排序过程与GTreord方法相比,调用DD方法的次数更少,能够更加高效地进行测试激励集重排序,求解速度有1~4个数量级的提高;此外,RTDD方法重排序后测试激励集用于故障诊断时,在保证同样诊断准确性情况下所需测试激励集个数更少.

参考文献

[1] LIU M, OUYANG D T, CAI S W, et al. Efficient zonal diagnosis with maximum satisfiability[J]. Science China Information Sciences, 2018, 61(11): 1-14.

[2] LIU M, OUYANG D T, ZHANG L M. A novel approach for improving quality of health state with difference degree in circuit diagnosis[J]. Applied Intelligence, 2018, 48(11): 4371-4381.

[3] 刘梦, 欧阳丹彤, 刘伯文, 等. 结合问题特征的分组式诊断方法[J]. 电子学报, 2018, 46(3): 589-594.
LIU M, OUYANG D T, LIU B W, et al. Grouped diagnosis approach using the feature of problem[J]. Acta Electronica Sinica, 2018, 46(3): 589-594. (in Chinese)

[4] WANG Y Y, LI R Z, ZHOU Y P, et al. A path cost-based GRASP for minimum independent dominating set problem [J]. Neural Computing and Applications, 2017, 28(1): 143-151.

[5] 欧阳丹彤, 刘伯文, 刘梦, 等. 结合电路结构基于分块的

诊断方法[J]. 电子学报, 2018, 46(7): 1571-1577.
OUYANG D T, LIU B W, LIU M, et al. A block-based diagnostic method combining with the circuit structure[J]. Acta Electronica Sinica, 2018, 46(7): 1571-1577. (in Chinese)

[6] ZHAN W F, EL-MALEH A. A new scheme of test data compression based on equal-Run-length coding (ERLC)[J]. Integration, 2012, 45(1): 91-98.

[7] 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)

[8] 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)

[9] POMERANZ I. OBO: An output-by-output scoring algorithm for fault diagnosis[C]//2014 IEEE Computer Society Annual Symposium on VLSI. Tampa, USA: IEEE, 2014: 314-319.

[10] POMERANZ I. Improving the accuracy of defect diagnosis by considering fewer tests[J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2014, 33(12): 2010-2014.

[11] POMERANZ I. Improving the accuracy of defect diagnosis by adding and removing tests[J]. IET Computers & Digital Techniques, 2015, 10(2): 47-53.

[12] BERNARDI P, GROSSO M, REBAUDENGO M, et al. A pattern ordering algorithm for reducing the size of fault dictionaries[C]//The 24th IEEE VLSI Test Symposium. NY, USA: IEEE, 2006: 386-391.

[13] BOLCHINI C, QUINTARELLI E, SALICE F, et al. A data mining approach to incremental adaptive functional diagnosis[C]//2013 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFTS). NY, USA: IEEE, 2013: 13-18.

[14] HUANG Y, CHENG W T, TAMARAPALLI N, et al. Diagnosis with limited failure information[C]//2006 IEEE International Test Conference. Santa Clara, CA, USA: IEEE, 2006: 1-10.

[15] XUE C, BLANTON R D. Test-set reordering for improving diagnosability[C]//2017 IEEE 35th VLSI Test Symposium (VTS). NY, USA: IEEE, 2017: 1-6.

[16] BODHE S, AMYEEN M E, POMERANZ I, et al. Diagnostic fail data minimization using an N-cover algorithm [J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2016, 24(3): 1198-1202.

[17] BODHE S, POMERANZ I, AMYEEN M E, et al. Reordering tests for efficient fail data collection and tester time reduction[J]. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 2017, 25(4): 1497-1505.

[18] BRGLEZ E, FUJIWARA H. A neutral netllst of 10 combinational benchmark circuits and a target translator in fortran[C]//Proceedings of IEEE International Symposium on Circuits and Systems. Kyoto, USA: IEEE, 1985: 695-698.

[19] BRGLEZ F, BRYAN D, KOZMINSKI K. Combinational profiles of sequential benchmark circuits[C]//IEEE International Symposium on Circuits and Systems. Portland, USA: IEEE, 1989: 1929-1934.

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. 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

刘　扬　男, 1994年出生于内蒙古自治区巴彦淖尔市. 2020年毕业于吉林大学软件学院并获硕士学位. 主要研究方向为基于模型诊断和测试诊断.
E-mail:ly2020@jlu.edu.cn

张立明(通信作者)　男, 1980年出生于吉林省榆树市. 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